AMSTRAD CPC 464 USER

95p
March 1985

Sound advice on Music

Dicing with Discs

Type-in Treats

Machine Code Mania

Plus

Loads of listings, Reviews and Stuff!

INSIDE Complete Software Checklist



WIN DK' Tronics new speech synth.


MICROPOWERMICROPOWERM
SUPERB SOFTWARE FOR THE AMSTRAD
SUPER-FAST LOADING TIME commodore 64
electron
B.B.C. MICRO
GHOULS
Run through the creepy mansion dodging ghostly ghouls and bouncing spiders. Leap over poison-smeared spikes, scamper along moving platforms and contracting floorboards, and use powerful springs to propel you onto overhanging ledges. Four screens.
Amstrad and Commodore versions £6.95
BBC and Electron versions £7.95
BBC and Commodore Disk price £9.95
Amstrad version
WATCH OUT FOR OUR NEW PACKAGING AND CATALOGUE
MICRO POWER
MICRO POWER LTD.,
NORTHWOOD HOUSE, NORTH STREET,
LEEDS LS7 2AA TEL: (0532) 458800
SELECTIVE BRANCHES OF BOOTS, CO-OP, MENZIES, W. H. SMITH, WOOLWORTHS AND ALL GOOD DEALERS.
AUTHORS! WE PAY 20% ROYALTIES!

SOFT-5004
Issue 4 March 1985
Distributed by SM Distribution
London SW9.

AMSTRAD CPC 464 USER

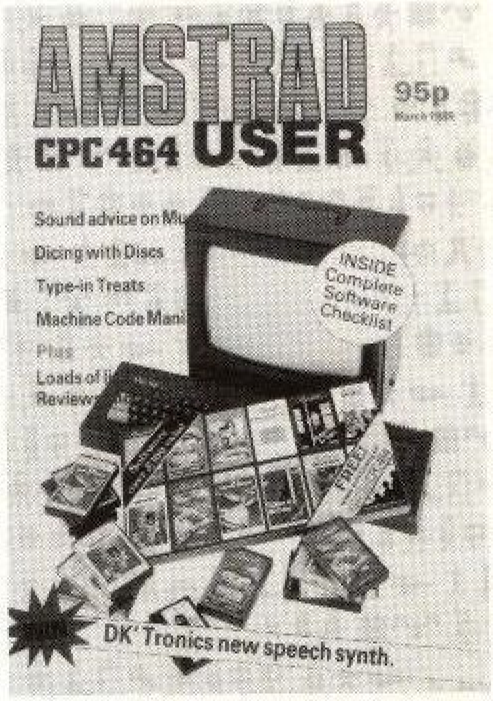
REGULARS

LANGUAGES

FEATURES

PROGRAMS

COMPETITIONS.



© Amstrad and Amsoft
No part of this publication may be reproduced without the permission of the copyright owner.
Please note that whilst every effort is made to ensure the accuracy of all features and listings herein, we cannot accept any liability whatsoever for any mistakes or misprints herein.
Contributions are welcomed from readers and professional journalists. If you want them returned, then please send a large SAE with all submissions. Please note the views and opinions expressed herein are not necessarily those of Amstrad, or Amsoft but represent the views of our many readers, owners, members and contributors.

Managing Editor
William Poel
Editor
Simon Rockman
Production Editor
Ivor Spital
Designer
John Alexander
Layout
Des Rackliff
Computer composition
Mike Dawsset & Miša Konstantinović
Photography
Chris Anstey
Consultant
Roland Perry
Advertisement Manager
Hilary Henshaw

CPC464 User, 169 Kings Road,
Brentwood, Essex, CM14 4EF
Telephone (0277)230222.

mmmmeggabusters from

fantasy SOFTWARE

THE PYRAMID & BACKPACKERS

now available on

AMSTRAD



fantasy SOFTWARE

BACKPACKERS PART 1 AND THE PYRAMID ARE AVAILABLE AT £7.50 EACH FROM ALL GOOD SOFTWARE RETAILERS OR FROM: FANTASY SOFTWARE, FAUCONBERG LODGE, 27A ST. GEORGES ROAD, CHELTENHAM, GLOS. GL50 3DT.

TRADE ENQUIRIES: TEL. 0242 583661

NEWS

A Ringside Seat



Daily Mail

MONDAY, DECEMBER 24, 1984

20p

Xmas computer ads spark a row

TYCOONS PUNCH IT OUT!

By JOHN HAMSHIRE

BRITAIN'S two top home computer tycoons, whose products will make up more than a million Christmas presents tomorrow, have been involved in an astonishing bar-room brawl.

One of the millionaire protagonists was Sir Clive

There can be few readers who have not yet heard the story of Christmas Cheer dished out in the 'Barron of Beef' hostelry in Cambridge.

Sir Clive Sinclair took exception to one of Acorns latest advertisments which quoted newspaper reports of the Sinclair Spectrums unreliability. When Sir Clive met Chris Curry, ex-Sinclair employee, and now one of the bosses of Acorn Computers, the two came to blows.

Never slow to turn a penny, Amsoft spotted the opportunity to produce a video game record of this titanic battle to be entitled "Business is War".

Had the prospective author, the talented Marcus Altman of Alligata (Blagger and many other stars of the small screen), not had an unfortunate loss of source code, we might have been previewing this epic herein.

The prospect of standing outside the launch of the Sinclair electric tricycle offering copies of the tape to passing journalists and generally making the most of the Sinclair press conference would have rounded off the whole affair very nicely. Rumours that the next Sinclair User show is to be held at the Empire Pool Wembley and promoted by Frank Warren are entirely unfounded.

What it's all about:

The famous sparring match in the Cambridge pub came about as a result of an advert published just before Christmas by Acorn, quoting several reference to a survey carried out by Business Decisions on behalf of a publishing operation in which Chris Curry (MD of Acorn) has 'a substantial shareholding'.

As you may have guessed, the survey was nice to Acorn and nasty to Sir Clive. Over 25% of Spectrums were returned to the stores as faulty in the warranty period, and only 5% BBC Model B computers - according to the survey, copies of which were not made available to those manufacturers cited.

And as for Amstrad? Well, much to the chagrin of some manufacturers who felt sure that a computer as new as the CPC464 would get a blasting, the CPC464 came out joint top at 5% with Acorn's BBC Model B and Electron.

Acorn's outrageous Christmas advertisement failed to acknowledge this fact: indeed, it was downright misleading suggesting that their two machines 'stood head and shoulders above the rest' and conveniently forgetting to remind readers:

1. That Acorn (and Sinclair for that matter) have been practising on the Britsh public for years anyway.
2. That the CPC464 system includes a tape deck and monitor - no other machine in the survey was so extensively 'exposed'.
3. That the CPC464 is obviously nicer than any other computer, so it was likely to be used far more anyway.....

W H Smith Back The Winner

As supplies have become more freely available, the retail giant W H Smith is at last able to stock the Amstrad computer system. The top 50 computer specialist stores will each have a complete system (including disc drive) on demonstration, and sales training currently being given to the staff should ensure good product knowledge.

'We're delighted to welcome W.H. Smith to the fold', commented sales supremo Dickie Mould,'We have now got as complete a distribution network as any computer manufacturer; and now we have to concentrate on improving software distribution to complement the hardware.'

Reports of poor software availability continue to hamper the CPC464, emphasising the important role being played by AMSOFT in ensuring that any owner has access to a full range of games, utilities and now CP/M programs.

Coleco Pull the Plug, MSX runs for the barn

In response to Coleco's departure from the computer manufacturing business, Amstrad Chairman Alan Sugar reminded the business of his prophecy at the launch of the CPC464, when he stated that the home computer business was shifting from the opportunist start-up operations towards mainstream electronic manufacturing operations.

Coleco (perhaps better known in the toy business) threw in the towel and jobbed out the remaining stocks of their Adam system in early January having resoundingly misjudged interest in a machine with no 80 column display, no low cost disc system, and no bundled monitor. Even a bundled printer failed to ignite any spark of enthusiasm. Maybe a free Cabbage Patch doll (also made by Coleco) with every purchase would have been more enthusiastically received.

Reports of sales of MSX machines around Christmas are, to say the least, unencouraging, and the immediate price cut prior to Christmas marks the most dramatic piece of panic yet witnessed in the business.

Database pulls a neat stroke

Thames TV Database production team are launching a program on Channel Four (Channel Four is like BBC2 with adverts) entitled, "Four Computer Buffs"

No not a typo, nor a reference to their average viewing figures, but a most laudable to provide thinking people with a computer program that attempts to behave somewhat like a 'Personal Computer World' on the small screen. In fact, most of PCW's production and contribution team seem to be involved somewhere along the line, this promises to be good wholesome fodder for Amstrad users.

Along with this breath of fresh airwaves, Database and Channel Four are planning to experiment by broadcasting computer software together with the test card in the morning starting around February 11th. So instead of Mantovani, you can waltz to the sound of Mega Byte and Parity Checkers as software for major micros - including the CPC464 - sprays from your TV set.

You can record onto most video recorders and then download onto a standard cassette recorder. Many VCRs have a direct audio output, and it's quite possible your local computer store or Comet service centre may have a lead to patch the output of the VCR into the input of the cassette.

Broadcast software on LBC has successfully been received simply by placing a microphone near the radio receiver - so this may also be a viable method - if you're at home between 10 and 12 in the morning. If you're not, then you had better start training your wife/mum/gran in the finer points of the technology straight away!

The first show in the series compares the CPC464 with the QL, and BBC Model B. But for software availability and a curious downmarking of the keyboard because it is a 'membrane' type, the CPC464 would have romped away with the honours, as it is, most people were pleasantly surprised by the proficiency of the CPC464 which actually beat the Model B on benchmarking (albeit two hundreds of a second overall) although not too surprised that the QL was 'seen to' by both machines sporting a quarter of the bits (according to the launch hype of the QL).

Software was assessed around last November (a lot's happened since). The keyboard marking was curious, since most other reviews have placed the CPC464 on a par with the BBC computer: as owners, you write in and tell them if you reckon the CPC464 keyboard is only marginally better than a QL.

Your scribe was able to be present to see fair play took place as far as possible - thanks to Michael Feldman, the producer - yet surprisingly neither Acorn nor Sinclair put up a representative to participate.

Amstrad Pulls It Off

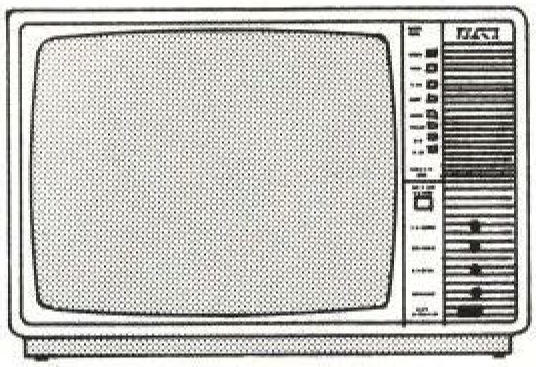
Another of Alan Sugar's launch prophecies came true at the turn of the year: sales Director, Dickie Mould, confirmed that Amstrad did indeed sell 200,000 CPC systems by the end of 1984, with a very heartening proportion going to export markets where demand has been well above expectation.

The CPC464 system is perceived very much in the Apple II market slot overseas, and with machines like the Model B at over £500 in countries like France, Amstrad is romping home as the machine 'most likely to' in 1985.



Alan Sugar - AMSTRAD's Chairman

Media Watch



In which the CPC464's TV appearances are chronicled. Reader input most welcome...

The CPC464 featured on Anglia TV's pre-Christmas buyers guide in the About Anglia program. David Kelly of Popular Computing Weekly played safe and recommended viewers to go where the software was, and the rather static QL on show appeared not to work at all. Ho hum, at least it looked sleek and meaningful.

CBTV continue to call their machine George (we tried to tell them it's called Arnold), and every now and again 'he' makes an entrance 'in vision'. On the Tuesday January 8th show he spoke for the first time as we fitted CBTV up with a DK'tronics speech synthesiser. Soon they'll be demanding he gets an Equity or NUJ card.

Channel 9 breakfast TV in Australia is a fascinating thing. It's hard to tell where the editorial stops and the adverts begin, but the man describing the CPC464 modestly as the greatest computer since the start of the universe can be forgiven for getting carried away just a little.

Micro Live on BBC2 continue to ignore the existance of the CPC464 with an studied indifference bordering on the outrageous, instead, filling their airtime with just the sort of assorted trivia that good media folk deem to be the meat and drink of computer enthusiasts. Must not laugh though, amidst much publicity and traipsing around computer stores, poor old Brian Jacks bought an Atari 800XL just before Jack Tramiel started slashing the price.

Former world Judo champion Brian Jacks versus computer industry Iron Man Jack Tramiel....? Now that really would be a rare old ding-dong! See you in the Barron of Beef!

Disc Software starts to arrive.



Amstrad seem more aware of the need for software support than anyone. It is great to see this being carried over to the area of disc software. The immediate job has been to transfer the best selling titles to disc. So far only sixteen games are avaliable but more and more titles are being added daily. As discs become more widespread it is likley that there will be some programs which actually use the facilities of the disc drive. This will mean that games can be up to 170K long and load in chunks.

In the near future Amsoft will be starting a scheme by which owners of tapes can return them in part-exchange for a disc version. For details you should write to the Amsoft sales department.

The packaging for the discs is very nice, it looks a lot like that used for the cassette inlay cards but has the curious label CPC464/664 along the side. The official Amstrad line is that CPC664 is the name the computer is sold under in some export markets. When a product is sold all over the world it is very difficult to find a name which is safe to use in every country so perhaps there is a ring of truth to this.

Free software.

The post-Christmas special offer for anyone buying a CPC464 is £100's worth of free software. This takes the form of a box of twelve programs: Xanagrams, Fruit Machine, Bridge-it, Easy Amsword, Roland on the Ropes, Animal Vegetable Mineral, Oh Mummy, Harrier Attack, The Galactic Plague, Roland in the caves, Sultans Maze and Timeman One. This provides a nice selection for most people.

This offer will benefit everyone, even those people who have had a computer for a while, because as more machines that are sold the more software houses will write for them. The twelve pack features on the front cover.

Amstrad User to go monthly.

You won't have to wait two months for your next issue of Amstrad User. The April edition will be on the newsstands from March 14th and on the second Thursday of every month from then on.

AMSTRAD CPC464 USER

EDITORIAL

The arguments over who won the Christmas battle are still going on with Acorn, Amstrad, Commodore and Sinclair all claiming to have had a good Christmas. Sinclair and Commodore are fighting over first and second positions whilst Acorn and Amstrad are both trying to sit in the number three seat. The feed-back from Acorn dealers is that they have not had a wonderful Christmas, perhaps because last year was very good due to the introduction of the Electron.

Amstrad succeeded in selling all the machines they had, the Christmas story may have been different if the supply had been even better. This has established the Amstrad as a major market force. Enterprise and MSX computers failed to make an impact. The third party software market is now alive and well as the directory in this issue shows. Even the third party hardware market is starting to bear fruit with the DK'Tronics speech synthesiser becomming avaliable. You can win one on page 45.

There have been two major areas of complaint in my post bag. They both concern the user group. The first complaint is that the binder which you get on joining the user group is too small for the magazine. There will be other things for the binder, such as errata and supplements to existing documentation. The binder is for storing these. The second complaint concerns deadlines. Many readers found that they could buy the magazine in the local news agent before their subscription copy arrived. This anomaly was caused by a shift of deadlines around Christmas and shouldn't happen again. I know I will look very foolish if you are reading this in the shop before your copy arrives.

Simon Rockman

Editor

Next Month.

Miss the April issue of AMSTRAD User and you will miss out on a lot. You won't get to see the comprehensive book reviews or the detailed analysis of the best joysticks for the CPC464. You won't be able to read the tutorials on how to write adventure programs or use music.

In next month's magazine there will be a machine code routine to allow you to add sprites to your programs.

If you have ever wanted more than twenty columns in mode 0 then you will want to type in the utillity which provides proportional print on the screen.

Beginners can look forward to Gareth Jeffersons article on how to use arrays.

Language freaks will enjoy the articles on Forth and Pascal.

If you are handy with a soldering iron then you won't want to miss the feature on how to add a battery backed up diary and real time clock to your computer.

The 3 inch discs are great but if you want to add a 5 $^{1}/_{4}$ inch unit then you should see next months magazine for how to do it.

All this plus loads of the usual stuff, programs to type in, pages and pages of software reviews, latest news, letters and hardware reviews.

The April issue is due out on March 14th. Don't miss it.

All articles mentioned here are in an advanced stage of preparation and are scheduled to appear in the next issue, however we cannot guarantee that they will all appear.

To make sure that you don't miss the next issue of AMSTRAD User fill in this form.

Dear Mr Newsagent.	My name is ....
Please keep a copy of AMSTRAD User every month.	My address is ....

the only choice

Kuma AMSTRAD CPC464 software



Holdfast | Gems of Stradus | Star Avengers | Galaxia | Music Composer



Logo | Database | ZEN Assembler | EASIVAT | Home Budget

An outstanding selection from Kuma's rapidly expanding range of Entertainment and Application Software for the Amstrad CPC 464 Micro-computer.



BOOK ● The Amstrad CPC 464 Explored.

This superb book is designed to let every CPC 464 user, at whatever level, get the most from his computer. After an introductory section on the special Basic features, the book looks in depth at the excellent sound and graphic facilities.

Now available from selected branches of Co-op, Granada, LASKYS and John Menzies

Kuma Computers Ltd., Unit 12, Horseshoe Park, Horseshoe Road, Pangbourne, Berks RG8 7JW.

Please send full catalogue on Amstrad CPC464 products.

Name ....

Address ....

....

.... Phone.............................

I own an Amstrad CPC 464 computer ☐

Trade Enquiries Phone 07357-4335

Visitors wishing to call at our Pangbourne Manufacturing and Distribution Centre are advised to phone 07357-4335 first for an early appointment.

You'll never dream a program could be this good!!!



PYJAMARAMA

Join Wally in his nightmare

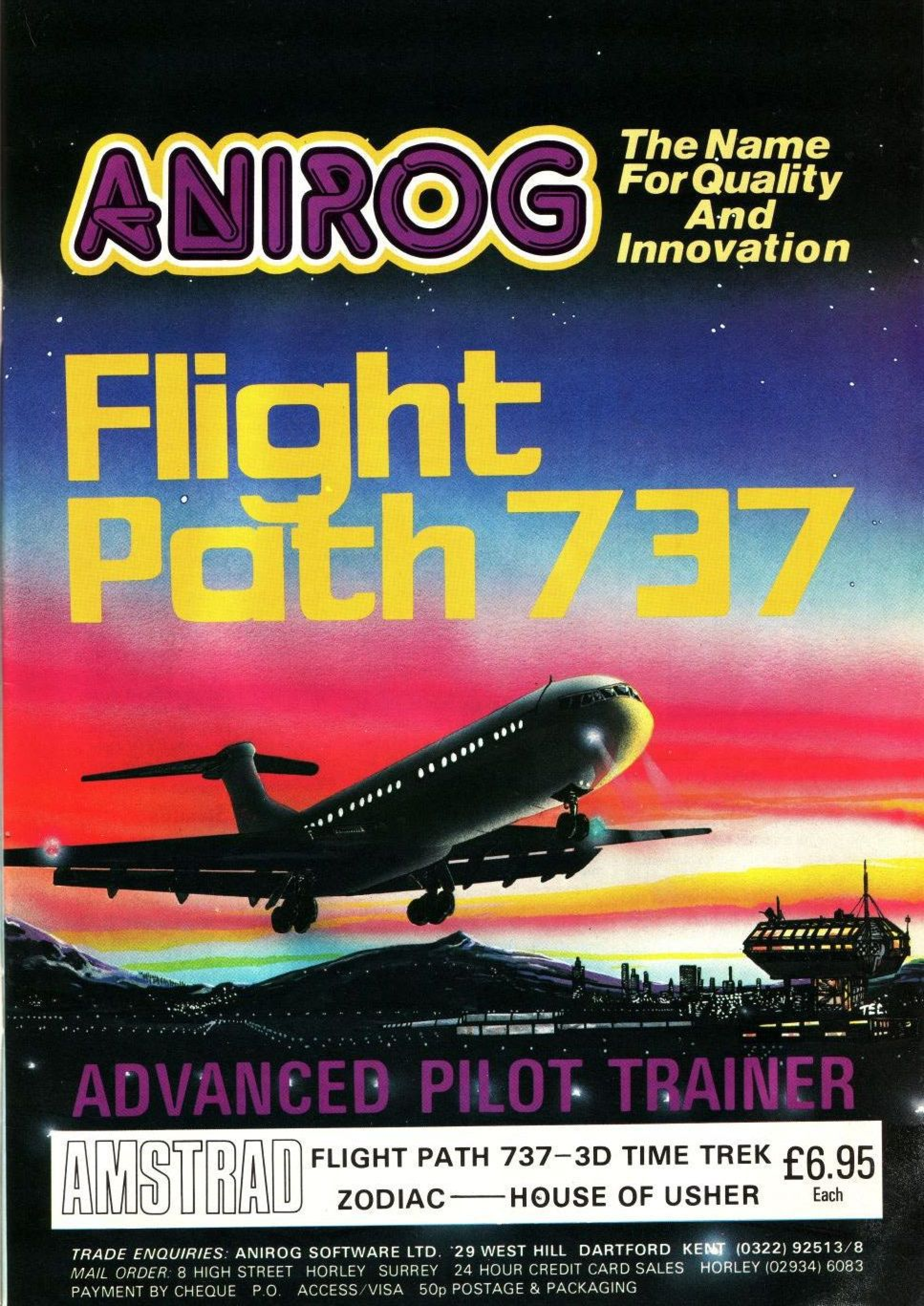
Sensational Software from

Commodore 64 – £7.95 **MIKRO-GEN** *Spectrum 48K – £6.95*

Amstrad CPC464 — £8.95

44 The Broadway, Bracknell. Tel: (0344) 427317

Available for Commodore 64 and Spectrum 48K


ANIROG
The Name For Quality And Innovation
Flight Path 737
ADVANCED PILOT TRAINER
AMSTRAD
FLIGHT PATH 737—3D TIME TREK
ZODIAC——HOUSE OF USHER
£6.95 Each
TRADE ENQUIRIES: ANIROG SOFTWARE LTD. 29 WEST HILL DARTFORD KENT (0322) 92513/8
MAIL ORDER: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT CARD SALES HORLEY (02934) 6083
PAYMENT BY CHEQUE P.O. ACCESS/VISA 50p POSTAGE & PACKAGING

Letters

Please bear in mind that the views expressed herein are not necessarily those of Amstrad or Amsoft. Be assured that all your views are given thorough consideration. This letters section is the CPC464 users' own forum.



Chiller

I do not expect you to publish this letter as its contents may well send a few shivers down the unsuspecting spines of new Amstrad owners! However, frustrated in my attempts to write a business package for the CPC464 by the lack of a machine on which to test routines, and having browsed through the second edition of the user mag which was packed with information on making the most of my micro, I thought I would vent some of my frustration on you.

You may wonder why I have no computer. Well, after only two months of reasonable use, it has gone back to its 'maker' to be repaired! The cause of this failure appears to be with the cassette unit. It refused to load any software at all, giving Read Error A or B and telling us to rewind the tape. A telephone call to Amstrad proved fruitless and so the machine was returned leaving me with an empty space on my desk and a pile of untested programs.

As we bought the CPC464 primarily for my husband's business it is extremely inconvenient to have it go down so soon and I begin to wonder if we were wise to switch from our original decision to purchase a BBC B. It also raises the question as to how reliable the disc drives will be! Obviously these are indispensable for business use but should we consider cutting our losses and switch to a tried and tested machine! Having invested in an expensive daisy wheel printer and the relevant interface, it's most likely that we shall persevere, crossing all fingers and toes, with the CPC464 providing that we do not have to wait too long for its return.

On the bright side, the computer itself appears to be most promising although I endorse all the criticism of the manual supplied with it.

One query that you may be able to answer as I'm not a hardware fanatic - Is it possible to interface two printers such that one can easily switch between the two without disconnecting any plugs etc? The reason for this is so that we can use a dot matrix for program listings, invoices etc. and a daisy wheel for business letters and direct mailing.

I sincerely hope that by the time this letter lands on your desk my CPC will have been returned and I will once more join the throng of happy Amstrad owners. Until then I remain,

'Yours slightly disgruntled',

B I Noyce, Emsworth, Hants.

CPC464 User: We can't imagine why you should think that we are intent upon suppressing all but the glowing praise: this would be a very sickly publication if all we chose to print was a eulogy for the CPC464! Well yes, things do go wrong, but I'm delighted to say that the CPC464 reliability factor is presently far better than the competition, and certainly streets ahead of 'that' computer you nearly chose when 'it' was only 4 months old. A lot of effort is going into service training and availability, and as you will see from the insurance scheme announced in our news section, the industry has unprecedented faith that the CPC464 is a good risk.

You are quite right that the disc is an essential upgrade for business. Amstrad intends to make them available at a price where everyone who is serious about their computing can afford one. The explosion of software that this policy will bring about will be very significant indeed.

Switching between printers is quite possible: the only catch is that the cost of the switch box can be rather higher than you might think. We have seen one such item in a well known computer supplies catalogue for just under £100! Maybe we can tempt some advertisers of such products to grace our pages before too much longer.

If you are going to plug/unplug printers all day, please use the Centronics plug end of the cable: it's far more robust than a circuit board edge connector.

Unstuck in Stradus

With reference to the reader stuck in the mazes of Gems of Stradus.

He has passed the rooms that hold the keys he requires which enable him to explore the adventure further. The pit he mentioned cannot be crossed until he has the plank, hidden deep within the maze behind the locked doors.

He also mentioned bugs, there is one (a cute one!). When warned that you are carrying too much, it appears that movement is restricted until something is dropped.

In fact if you type 'LIST' then make a move, it is bypassed. You then get a warning again but just keep on as above, until you arrive within more familiar surroundings, where the item can now be dropped.

Also, once past the bugs the stilts are

useless, so they should be dropped. Readers should also leave the snake alone until the safe has been opened.

I hope these tips prove useful to other intrepid adventurers.

J Marshallsea, Oldham

Unstuck again

Many thanks for issue 2 of the 'CPC464 User' -improved out of all recognition -when are you going to start marketing a binder for it?

I am one of the many who no doubt are inundating your mail box with advice for RJS of Sheffield (Stuck in Stradus, Issue 2, CPC 464 User).

Suffice it to say that provided he has collected the eight keys he should have encountered in reaching the pool room (opening all closed doors but the one next to the sword room on the way), the locked yellow door opposite the door to the first open darkroom will open. He should also have picked up two essential items omitted from his list on the way there.

More than 50 other rooms lie beyond that yellow door, yielding, eventually all the items (and more) that RJS is looking for and will need to outwit the mighty Am. The best of luck to him!

Incidentally, I believe there may well be an unexpected 'feature' - even though its friendly! If you get to the timelock vault and haven't got the first timelock number, try entering '000' as the first number. It seems to work on my copy! This can save six moves if you're trying for an all time low score.

As for the CPC464 itself, it continues to amaze and delight me, my family and friends with its sheer brilliant excellence! The more you use it, the better it gets!

Now for a rare instance - a CPC464 failure! My brother bought one from Boots in July of this year and in September the cassette motor played up, resulting in an embarrassing spate of 'read error b's. Boots immediately and without fuss exchanged his computer for a new one, consoling him with the information that it was the first example (at that branch anyway) which they had had returned for any reason whatsoever - and they had sold lots.

It must be a great relief, to owners and to computer salesmen alike, that Amstrad, at least, seems to be resisting the temptation to use its customers as (unpaid) quality control inspectors. (Having said this, it is now almost certain, I suppose, that my CPC464 will immediately malfunction!).

Brian Cass, Derby

CPC464 User: Computing would be fine if it wasn't for the mechanical bits. We are not aware of any IC failures apart from a couple of shaky RAM devices that lead to a spot of amnesia. The CPC464 cassette still seems to be very much more reliable than any machine with an external cassette requirement, so there's no panic setting in!

Although discs have mechanical elements, we are delighted with the reliability of all the discs we have been using in field trials and cannot stress how much more enjoyable and less frustrating computing with discs can be.

Beware leaving the cassette rewinding against the end stop. We've all done it, and too much leads to slippage of the pulleys and bands as they shine themselves up against a static cassette spindle.

Stradus: The last word

After many attempts at Gems of Stradus I got through! The screen said 250 moves was excellent, no bugs qed.

To help RJS, Sheffield, through the maze find enclosed my map and key. Also a list of items and their counterparts.

However, to provide a simple answer to his question:-

To light your way through the dark room you require a lamp, matchbox and matches. Also the dog needs a bone to keep him quiet while you slip past to get the plank. Play on ...

C Byrne, London

P.S. I read my papers, do you?

CPC464 User: Mr Byrne provided us with a fully commented map and list of objects and their uses, but you wouldn't want us to give the game away, or would you? Persons providing a doctor's certificate to the effect that failure to complete this adventure is proving injurious to health may just get us to part with a copy.

A Happy Wanderer reports:

I have just read issue two of CPC464 user, its pretty good but it's very biased.

Anyway enough of that, let's get down to business. The reason I am writing is that I have found all 53 screens of that excellent game ROLAND IN TIME. I have a full list of all the sheets below and some game tips as some of the sheets are hidden, crafty that eh! The game sheets that are hidden are:

Time Zone 7
The Crumbling City
The Bakerloo Line
Death of a Landmark
Time Zone 3
Stepping Out

To get to the hidden sheets you must first enter Time Zone 7 then go out of the 'phone box. Now turn around and jump left, you should now be on top of the 'phone box: jump left again and you will be on a hidden opening, now walk left and you will be in a hidden room. Hope you find this interesting.

M Murray, Rayleigh

Roland in Time

1. Down at Dollis Brook
2. Ye Olde Oake Tree
3. Pterodactyl Nest
4. The Ice Cavern
5. The Ice Maze
6. Amity Littoral
7. Jacob's Ladder
8. Shelob's Lair
9. Beneath Ye Oake
10. The Roman Fort
11. Oh Mummy
12. Death on the Nile
13. Temple of the Gods
14. Njorl's Dragonboat
15. Valhalla
16. Where Eagles Dare
17. Njorl's Castle
18. Romsdalsfjord
19. Njorl's Cavern
20. Seen my Ballista?
21. Above the Ramparts
22. Foot of the Tower
23. Castle Gateway

24. The Guardroom
25. The Turret
26. Stepping Out
27. Jenny Goes Spinning
28. Trouble At Mill
29. Roland Goes Digging
30. Devil's Bridge
31. Missile Silo #47
32. Stansted Airport
33. Sainsco Supermarket
34. At the D.H.S.S.
35. The Computer Shop
36. The Production Line
37. After the Holocaust
38. The Crumbling City
39. The Bakerloo Line
40. Death of a Landmark
41. Heli Hunter Station
42. No Mans Land
43. Home Tree Base
44. Bough Wow
45. Tree House
46. Admin Complex 8
47. West of Dome 9
48. Alien's Retreat
49. The Two Towers
50. Dome 9
51. East of Dome 9
52. Dome 9 Spaceport
53. The End of Time

CPC464 User: What can we say? Gem had better hurry up with the sequel!

Unimpressed

I am an enthusiastic CPC464 user but reading last month's magazine, I was disappointed to see that in common with most other computer manufacturer's, you have decided that users and readers will be men. Why do we have to see a naked woman illustrated advertising the hardware section?

How about some articles and software for girls and women - who in general aren't fascinated by games showing how to randomly bomb buildings.

Paulene Morgan, Ballymena, Co. Antrim

CPC464 User: OK then ladies: how about it? The only specifically feminine software we've seen has involved rushing around a kitchen, stopping pans boiling over, and various knitting pattern generators. This might raise an even greater howl of protest.

Can anyone get the GLC to stump up a grant for some non-sexist software before that body is summarily deposed?

Prime PASCAL

With reference to issue number 2 of your magazine and the article 'HISOFT PASCAL' page 54, Example 2, a prime number program.

The following program achieves the same result in thirty one seconds.

```
10  dim s%(5000):n%=65:e%=4999:print "2";
20  for p%=3 to 31 step 2:if s%(p%)=n%
    then 100
30  print p%;
40  for I%=p%+p%+p% to e% step
    p%+p%:s%(I%)=n%:next
100 next
110 for p%=33 to e% step 2
120 if s%(p%)<>n% then print p%
130 next
```

E Simpson, York

CPC464 User: 31 seconds, eh? Unless you know better.....?

Down with Sinclair!

First of all I disagree with what G S Sinclair wrote in his letter, issue 2 Oct/November stating that the CPC 464's manual would be unhelpful to a complete beginner, well as I purchased the CPC 464 for my son who is eight years old and a beginner, the manual in my view is ideal for learning Arnold's potential, perhaps he would like Amsoft to supply an O-level standard manual with Arnold, and leave all beginners in the dark.

For the more advanced operator you give an option to purchase more advanced manuals.

Your correspondent also states that he has mistaken his on/off switch for his volume control and ended up switching off. Well as I have said my son is eight, he has had Arnold for nearly three months and has yet to make this mistake. Is the membership fee a bit steep? Well I'm not a rich man, but I do know and have faith in Amsoft that we will get good service for the money.

Lastly, you might ask why choose the CPC464 as a first computer, well it was very good value with a colour monitor, is smart looking and has a very good keyboard, also I have dealt with Amstrad and had very good service before and after. You seem concerned that your customers get a good deal.

Yours sincerely (FAN OF ARNOLD) Mr G Duffy and Leigh Duffy (Member), Harlow, Essex.

CPC464 User: The battle rages! The user guide supplied with the machine is the centre of much controversy. First timers seem to like it, old hands feel it's not enough. Maybe we could get another couple of hundred pages for the next one: and best of all, why don't some of the users write the chapters they would like to see themselves? We may even pay them for it!

Slow, slow, quick, quick

Firstly let me congratulate you on a very fine User's magazine and group. What a pity other manufacturers do not do the same With some thoughts for Amsoft, while ARNOLD allows us to have a fast and slow loading/saving speeds. Why does AMSOFT insist on having a slow loading version of the game on each side of the cassette. For people with disc drives this is no problem but for the poorer people like me who cannot afford a disc drive it can take eons for a game to load off cassette, why cannot AMSOFT have a slow loading version on one side and have a FAST loading version on the other side of the cassette like Nemesis software does. A lot of us poorer people would be very grateful for this little service.

While on the subject of disc drives for ARNOLD, I agree with your comments in the second news letter about transferring tape programs to disc. For the simple price of the disc, it is not a bad offer really, as a lot of manufacturers do not even offer this simple service. Well done AMSOFT.

As for this when I can afford a disc drive I will want my programs transferred, but I do not have the original boxes but I do have the original cassette cases and inlays - will this be sufficient proof of purchase as I do not have room for the boxes?

As for Arnold itself I find the escape key keeps getting pressed instead the '1' key. Also there seems to be great difficulty in peeking the screen to see if a certain character is present at a certain location. Has some clever clogs over-

come this? If so I know I would like to hear more. As regards the manuals, they are very good, but I think the firmware book should be included with the machine (I aint got one yet).

Yours sincerely, a friend of Arnold

G Tomlinson, Stapleford, Nottingham

CPC464 User: Cassette duplication speed is being experimented with at present: American football comes with fast and slow sides, and we will see from this what the results are like. We hear of so little loading trouble with the 1K speed that it's been a safety first policy.

The final policy on disc upgrades has yet to be fixed. The part exchange would be for original AMSOFT cassettes: not anyone elses, not pirated copies! The cardboard boxes are being dropped now that we have established the fact the software exists, and there should be enough about that dealers don't have to be presented with a large cardboard outer to emphasise the point.

On the Mat

May I endorse Mr Aston's reservations about the instruction booklet for Codename Mat? Not only does it substantially understate the extremely high quality of the game, it adds an adventure aspect which the author cannot have intended by failing to mention several vital control keys.

Anyone still thrashing around space using 4 to reach 30% speed as advised by the instructions may still be wondering how to stop or speed up. You soon discover that 3 reduces speed fairly rapidly to 0, but may wonder how to obtain a sufficiently delicate touch for docking. The keys you need are 1 and 2. 1 reduces speed by 1 or 2% each time it is pressed. 2 effects a similar increase. 5 increases speed to 100%. Only the 'QWERTY' keys though; the more accessible numeric keypad is disabled.

If, like me, you cannot be confident of being able to devote over an hour to saving the universe without interruption from the doorbell or telephone, it helps to know that H will halt the game and keep it in suspense until another key is pressed.

Finally, fast though Locomotive Basic may be, it is always worth avoiding unnecessary delays. If REM statements are offset from the subroutine line by one, the program does not take the time to go there to find that it need not have done so!

G J Edwards, Guiseley, West Yorkshire.

Third time lucky

After reading the second issue of CPC464 User I thought I would write in and say how much I like your magazine. The CPC464 is my third computer (I have owned a VIC 20 and a 48K Spectrum). There is one problem though, the price of the software. Why does it cost so much, I mean £8.95 is a bit pricey. You have some companies that sell software for £1.99 like Mastertronics and are still going strong, so why can't you sell your software for a lower price than £8.95, say for £5.50, then more people would be able to afford to buy the excellent software.

This program sorts numbers in to order.

```
10   Rem Number Sort
20   CLS
30   PRINT:PRINT
40   INPUT " How many numbers do you
     want to sort,";N
50   DIM A(N)
60   FOR X = 1 to N
70   LOCATE 1,2: PRINT "Next Number"
80   LOCATE 36,2: PRINT 0+X
90   INPUT A(X)
100  LOCATE 14,2: PRINT A(X)
110  IF INKEY$ = "" THEN 110
120  CLS
130  NEXT X
140  LOCATE 16,2: PRINT "Sorting"
150  S=0
160  FOR B = 1 TO N-1
170  IF A(B) > A(B+1) THEN GOSUB 250
180  NEXT B
190  IF S = 1 THEN 150
200  CLS: LOCATE 17,2:PRINT "Sorted"
210  FOR X = 1 TO N
220  PRINT " ";X;"=";A(X)
230  NEXT X
240  GOTO 300
250  T=A(B)
260  A(B) = A(B+1)
270  A(B+1) = T
280  S=1
290  RETURN
300  IF INKEY$ = "" THEN 300
310  CLS
320  END
```

Sorry this was not printed out on a printer. I hope to have enough money by Christmas to buy one so that next time I write, it will be done on a printer. I am about to start a course in computer programming and hope to have enough money by the time the third issue of this magazine is out, to buy the DEV-PAC assembler and The Concise Firmware Specification (Soft 158) to help me in my studies.

Mr S Turner, Belvedere, Kent.

CPC464 User: Thanks for the sort routine. Please send a tape if you can't send a listing: we can list ourselves: the number of mistakes a copy typist makes when trying to enter the above program from a handwritten listing (which was really quite legible to a programmer with some experience) is quite mind boggling: we hope this one is correct!

You already know our view on software from an earlier letter.

Shoot the typist

I am writing, firstly to congratulate you on an improvement on issue 1 with issue 2 and secondly with some useful tips.

On line 3000 of the date program (p38, issue 2)

```
M$ = MID$9M$...
```

the 9 should be a bracket (.

Pontoon Revisited can be further enchanced by the following. Firstly change lines 70 and 130 to 3 and 4. Then add the following lines:-

```
2 pounds = 100
211 WINDOW #1,1,40,20,25
212 PRINT #1, "Place your bet,
    you have £ "pounds
213 INPUT #1, q
214 IF q > pounds then print #1,
    "You only have £"pounds:
    FOR x=1 to 500: NEXT:
CLS#1:GOTO 212
215 pounds = pounds-q
216 CLS #1
217 PRINT #1, "Amount bet £"q
511 IF pounds < = 0 then 520
512 FOR x=1 to 4000: NEXT: GOTO 20
681 pounds = pounds + (2 * q)
682 GOTO 511
721 IF pounds <0 THEN GOTO 511
722 GOTO 20
```

Well done once again for an excellent magazine.

Mr P Meade, Chippenham

What's in a name

One article in CPC464 User 2 starts 'Whatever else you do, don't call the CPC464 'the Amstrad''. For better or worse, this habit is nearly universal and it's probably too late to stop it. Read almost any magazine!

I spoke with several dealers at the recent PCW show, and they were unanimous about wanting our games for the Amstrad labelled as such. The logic was that 'CPC464' is not the most memorable of names and would confuse customers.

Our Memotech games, on the other hand, are all labelled 'MTX' (the correct name of this micro) and dealers hated this. We're changing to 'Memotech' as soon as possible.

The moral is that it doesn't pay to be too pedantic. And if you've got a great name like 'Amstrad', why not flaunt it!

Best Wishes

Pete Austin, Partner, Level 9 Computing, High Wycombe, Bucks.

CPC464 User: I think we have just about given up by now. Short of transportation for offenders, there's little left for us to do.

Don't say that you weren't warned when there's another 10 computer products with Amstrad on the name plate. And don't forget that CPC464 is universal - some countries market the CPC464 under different brand names, but the model type number is the same everywhere.

Ambiguity

If one looked at the best selling micro-computers, it is apparent that the successful computers are either 'Games Machines' or 'School Machines'. On the games side, hopefully, you are taking care of matters, however, on the school side, the Amstrad is virtually unkown. This is a sad fact and is an area, with the CPC464's value-for-money integrated system that it can do well.

The best way to open the educationalist's eye is to hold a nationwide competition, with twenty 464's being the major prize. You should advertise in magazines and if possible persuade a popular children's television programme to work in conjunction with Amstrad. Interlaced in the competition, you should emphasise the 464's sophistication and of course its value for money integrated system.

After the competition, you will find the Amstrad will meet with a lot less ambiguity in schools and colleges.

K Lussain, Barking

CPC464 User: What the CPC464 needs is a convenient means of easing teachers out of the embarrassing position they find themselves in when advising their superiors that there is a computer now available which costs rather less than half the one they have been extolling to date.

The pupils and parents might lead the way by insisting schools use computers which may be economically reproduced in the pupils' own homes, without taking out a second mortgage.

As for a popular TV children's TV program, watch out for CBTV on Thames at 4.45 on Tuesdays. Possibly nationally networked, although the times may vary.



Software Reviews

Pyjamarama

Wally has fallen asleep and finds himself in a nightmare where everything appears to be larger than life. In order to wake him up you must find the alarm clock and wind it up. Sounds easy doesn't it? However, many objects are required to achieve this task and just when you think that you are getting somewhere, you find that you havn't got the right key for the next door.

Just to make life even more difficult, Wally has a limited amount of 'snooze energy' which decreases each time he clumsily bumps into something or even more clumsily lets something bump into him -remember Wally really is a wally! All is not lost however, since Wally's energy can be replenished by consuming any of the items of food scattered throughout the house. But beware, losing Wally's snooze energy 3 times results in the end of the game.



Throughout the game you will find that parts of the nightmare repeat themselves, repeat themselves.... and the order in which you must collect and use the objects becomes most confusing, if you manage to sort it all out then let us know at Amsoft how you did it!

I particulary liked the clear graphics, however better colours could have been chosen but I suppose a certain amount of artist's licence is allowed since after all, it is Wally's dream world.... Look out for the boxing glove and the mysterious hand which appears from the floor!!

A catchy tune accompanies the demonstration mode of the game which fortunately switches itself off when the game starts, since after a while, the continuous noise gets rather tedious, although in desperation you could always turn down the volume control on the CPC464!

Definitely a game worth considering if only to admire the graphics and the touch of humour which surrounds the game.



AUTHOR:	MIKRO-GEN
FORMAT:	CASSETTE
PRICE:	£8.95
GRAPHICS:	*****
PLAYABILITY:	****
ADDICTIVENESS:	***
OVERALL:	****

Dragon's Gold

On first impression this game appears to be very easy but I found that even on Level 1 it was difficult to achieve success. Set within the confines of an old castle, your task is to run through each of the awaiting



6 rooms, pick up some treasure in the last and return it to the first where you have left your empty chest.

It really sounds quite simple but you must take into

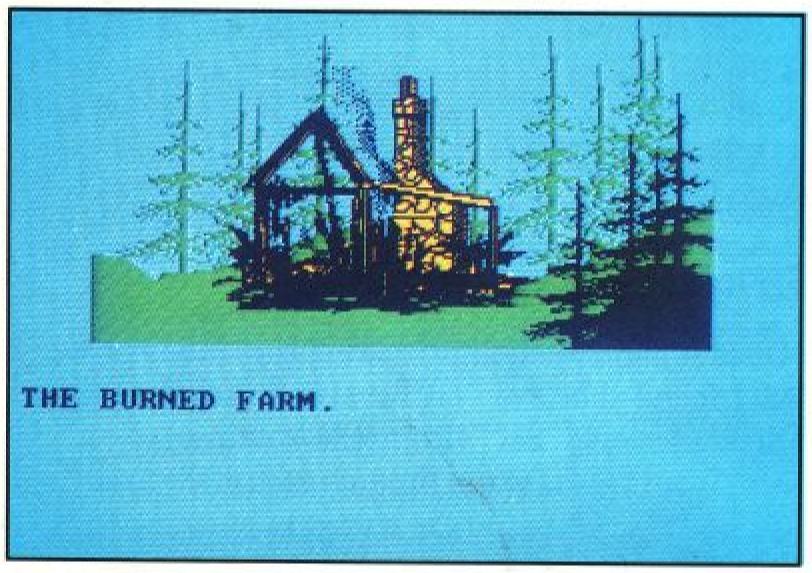
consideration the many hazards which await you as you try and dash through each of the rooms. Probably the most difficult of these is room number 5 'the corridor', which with its moving walls, takes a lot of time and patience to conquer. You are of course armed with a 'magic missile spell' but this won't help you against moving walls!

Simple graphics together with the simple task in hand give this game initial interest but I would think that once you have mastered each of the rooms, the game would soon lose its appeal.

AUTHOR:	AMSOFT/ROMIK
FORMAT:	CASSETTE
PRICE:	£8.95
GRAPHICS:	**
PLAYABILITY:	***
ADDICTIVENESS:	**
OVERALL:	**

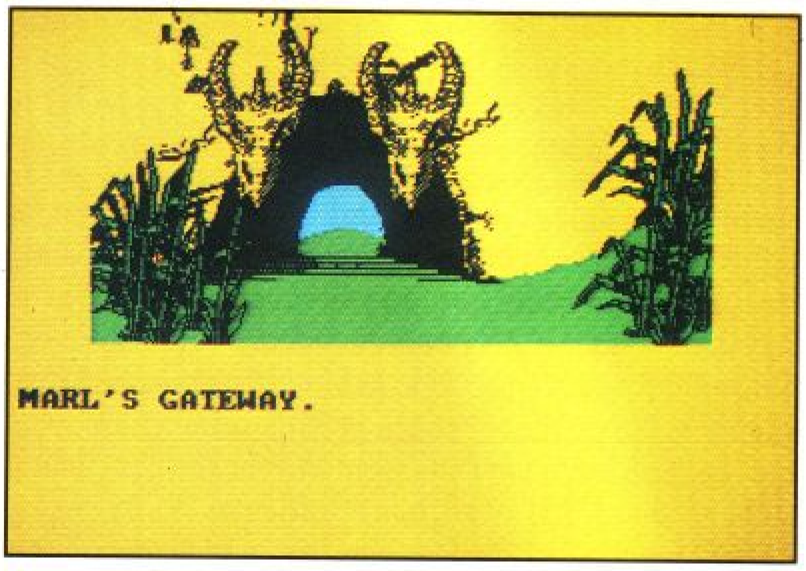
Forest at World's End.

Graphic adventures have been around for a couple of years now. The CPC464 is well suited to this kind of game; it has plenty of memory and very good graphics.



The major critism of graphic adventures is that they add nothing to the game. Hardened adventure players would rather have more text and faster responses. With Forest at World's End there is a flaw in the argument. The graphics add atmosphere to the game. It is not surprising that it takes ages for the program to load. Not all the locations have an associated picture, but those that do are great. The graphics are the work of Terry Greer. He has really got to grips with the Amstrad and has produced the best graphics seen on the machine. The routines to draw the pictures are impressively fast and the scenes depicted are very apt. Most modern adventures will accept long sentences, the routine which does this is known as a parser. How good the parser is affects the playability of the game.

However 'Forest' is quite limited in both its parser and vocabulary. The command to get into the boat is



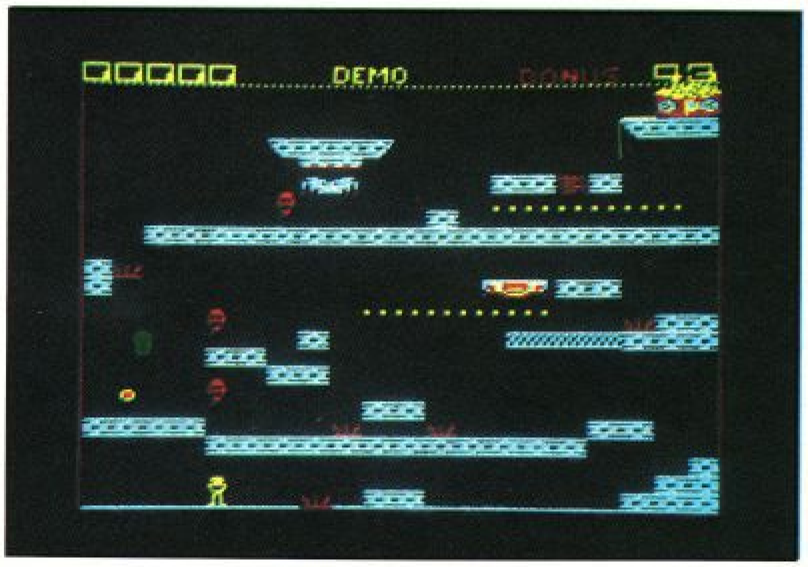
'climb into boat'. 'Enter boat', 'go boat' and 'jump into boat' won't work. This means that a lot of time is spent guessing what commands the program knows. A large vocablary takes up a lot of memory and, to be fair there is a lot packed into this game. As with all good adventures there is a save game feature but the lack of a score option leaves you puzzled as to how well you are going. 'Forest at World's End' is not a particularly difficult adventure with none of the nasty traits which put the novice off the genre. An ideal first adventure.

AUTHOR	INTERCEPTOR SOFTWARE
FORMAT	CASSETTE
PRICE	£6.00
GRAPHICS	*****
PLAYABILITY	****
ADDICTIVENESS	***
OVERALL	****

Ghouls

This game is, as its name suggests, about ghosts, spooks and ghouls.

It is similar in style to the now well known Manic



Miner and Roland in Time, and although the graphics aren't as good, it is very enjoyable, challenging and difficult.

You, the Pac-Man style player, have to run along jumping up to high platforms or falling to lower platforms.

To make it a bit harder there is a ghost or three to prevent you from reaching your house (at the top of the screen), these ghosts start above you somewhere, then randomly, and slowly (at first) make their way down towards you until they reach you, in which case you lose a life. You start with five, but don't be disheartened, you do have a chance; only slight though. Halfway up the screen is a gold ring and if you reach this before you get hit by a ghost, you destroy all the ghosts on the screen, and so can continue at your leisure, although you shouldn't take too long or your time (which starts at 99) runs out and, you guessed it, you lose a life.

If you are good enough, you can get your name on the high score table which holds up to eight.

This is a very good game and should keep you spooked for many hours.

AUTHOR	MICROPOWER
FORMAT	CASSETTE
PRICE	£6.95
GRAPHICS	****
PLAYABILITY	**
ADDICTIVENESS	***
OVERALL	***

Classic Adventure

Adventure games have been around for a very long time, they owe their existance to role-playing board games. Classic Adventure from Abersoft is based on the very first adventure game written in America by a couple of university students called Crowther and Woods. The original written in the language FORTRAN ran on a mainframe computer. It was so popular that it has been copied the world over and most self-respecting computers have some form of this game. The Amstrad is pretty lucky because there are two. The Amsoft version, called Classic Adventure and the Level 9 version called Collosal Adventure.

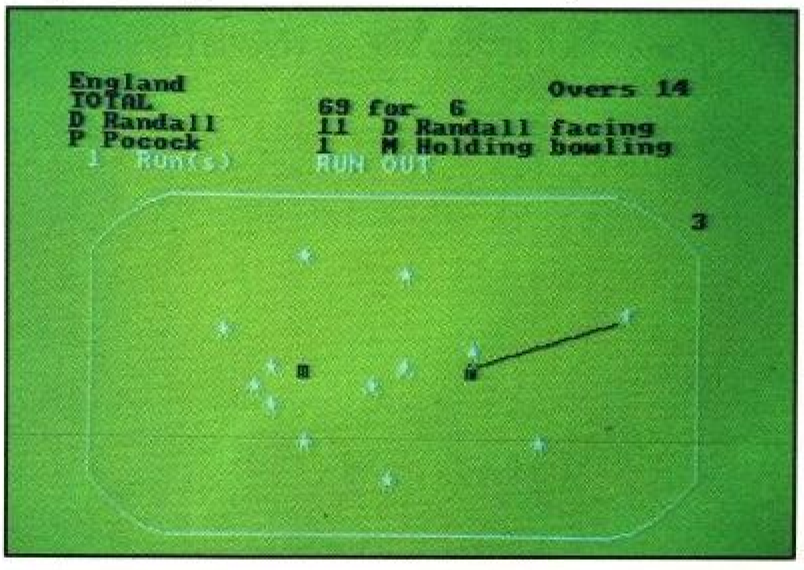
Most adventure players have cut their teeth on a version of the original game, so if you get stuck it is usually possible to find someone to drop a hint. The Amsoft game adheres closely to the shape and descriptions of the other versions. There are no really nasty puzzles. In Classic Adventure you start from a well house near a small stream. Around you there are various objects, in this game every object has a use so it is worthwhile picking up all the objects you can carry.

You are told that there is a cave nearby, reputed to be dangerous but full of wonderful treasures. The initial task is to find this cave and then plunder it. Hazards inside include an ogre, a snake, a particularly puny dragon and hordes of nasty dwarves. The treasures include gold, jewels, an emerald egg and a platinum pyramid. To get all the treasures you need a devious mind and to use the odd bit of magic. Classic Adventure is getting a bit dated now, you are limited to two word entry, there are no graphics or intelligent opponents. However it is a part of computer foklore, without having played classic adventure you will miss out on knowing the meaning of XYZZY,Plugh and Y2. This is one game to include in the collection, it is great fun to play in a crowd and allows you to escape from the real world of bugs, syntax errors and school.

AUTHOR	AMSOFT/Abersoft
FORMAT	Cassette
PRICE	£8.95
GRAPHICS	N/A
PLAYABILITY	**
ADDICTIVENESS	****
OVERALL	***

Test Match

This game is run on the CPC464 in a similar way to American Football reviewed in the last issue. It assumes you are familiar with the basic rules of cricket. The instruction pages go to some length to explain how the program processes the rules and in particular the



way in which the outcome of each ball is determined taking into account the abilities of the batsman and bowler. The user can either select and name the players in two teams or more simply use those already in the computer memory, England and the West Indies.

The game consists of 2 innings for each team or if not completed within 400 overs a draw is declared ,so be

prepared to make a long evening of this game. The pitch and players are displayed graphically on the screen, if the batsman hits the ball the fielder nearest will flash and you are required to decide whether or not to take a run, the path of the ball being traced on the screen as the fielder throws forward the wicket. At the end of each over a complete score card is displayed, and a new bowler chosen for the next over.

Most eventualities are catered for, such as boundaries and wides, and the program will not allow the same bowler on consecutive overs.

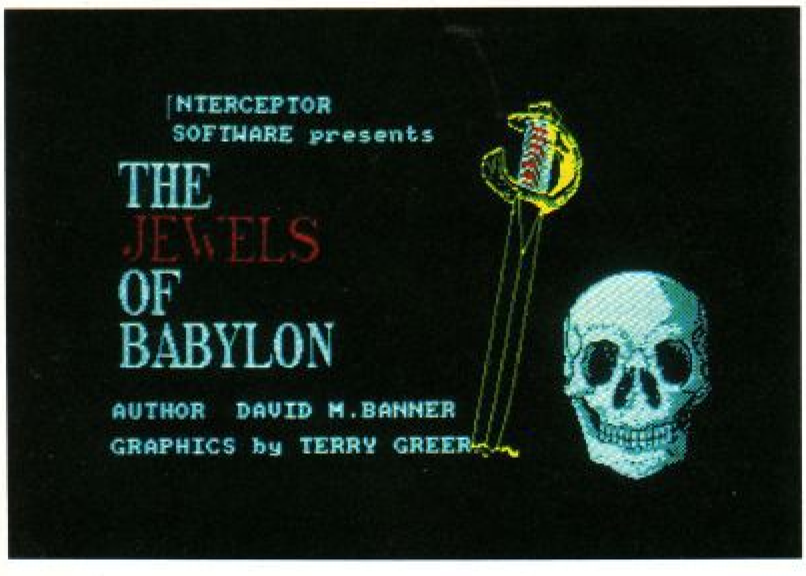
A sound adaptation of cricket to run on the CPC464, but I feel you need to be a keen cricket fan to find lasting appeal in this game.

AUTHOR	CRL
FORMAT	CASSETTE
PRICE	£6.95
GRAPHICS	****
PLAYABILITY	***
ADDICTIVENESS	**
OVERALL	***

Jewels of Babylon

This is another adventure with graphics production from Banner/Greer of Interceptor Software fame. With graphics of the same high quality as 'The Forest at World's End' but not limited to a small window in the centre. Loading is quick, as much of the game is loaded as one large block which sounds unnerving the first time you hear it.

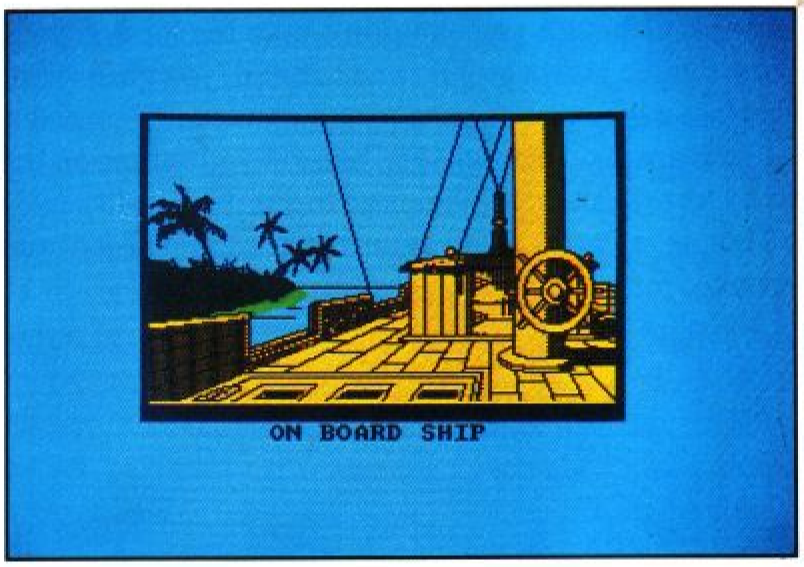
This adventure puts you in the setting of a 19th century sailing ship which came off badly in a tangle with some pirates. You are the sole survivor of the attack



and venture forth to recover the booty from the pirates; namely The Jewels (a wedding present from Queen Victoria no less!). On the island you will meet such charming creatures as crabs, crocodiles, spiders, water snakes and an octopus. All creatures are supplied in the large economy size folks!

It is very easy to get lost on this island and spend ages running around in thickets, dense jungle, swamps and the like. Eventually one encounters some obstacle (lion, crocodile etc) standing there. After dealing with the problem you notice an extra exit availible in that location which is usually worth following up.

The swearword and curse routine is very limited and this can provide simple amusement as when faced with a verb it is not familiar with, it replies 'I don't know how to .....' On the more serious side it does not recognise many (to me) obvious words that it has used to describe a location. For instance, telling it to, 'go down' at the initial location on the ship, gets the reply 'you can't', after it has said that down below there is



a rowing boat. You also have to type, 'climb into boat'; 'climb in boat' apparently will not do.

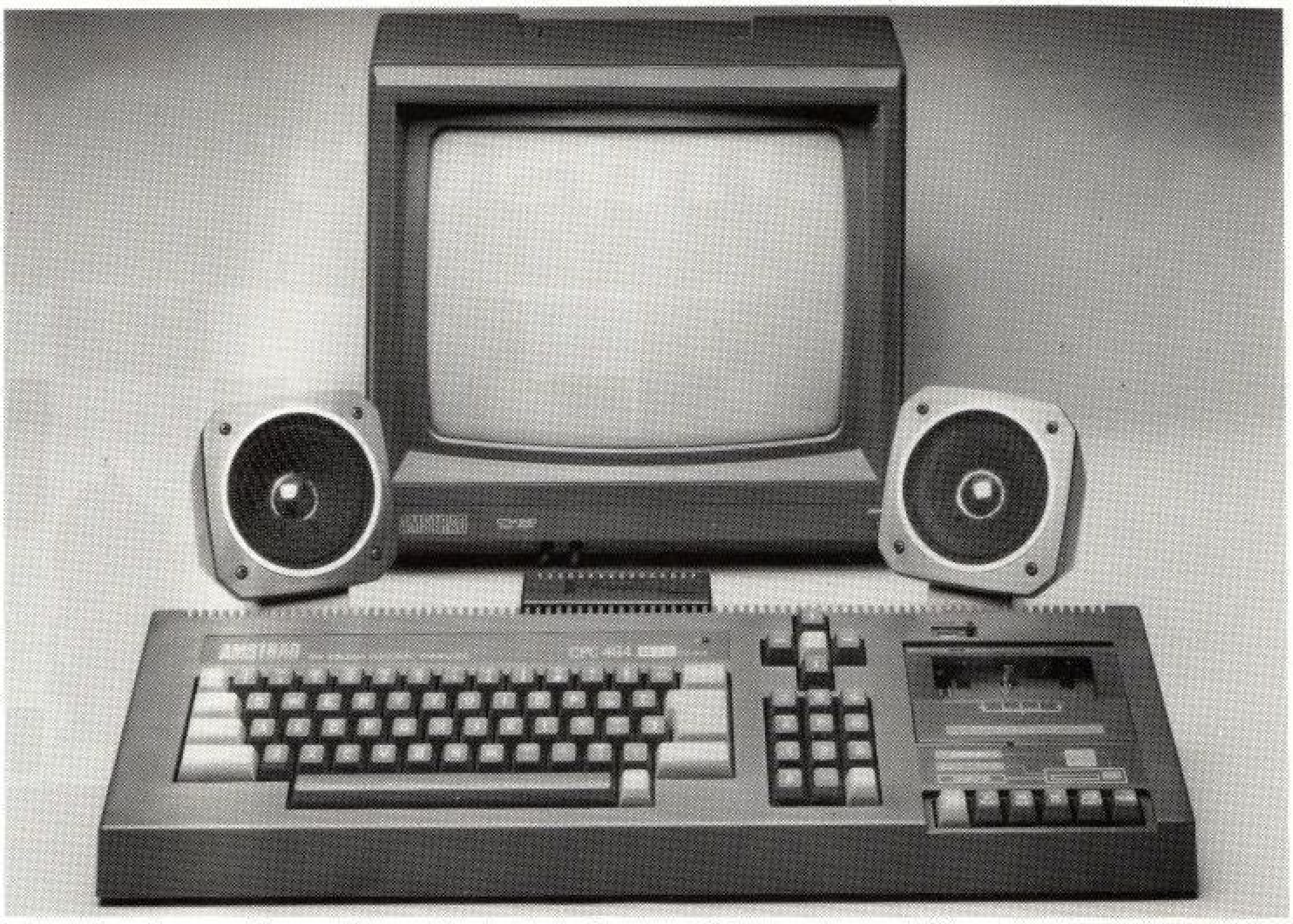
The graphics, pretty though they are, can get a trifle repetitive as it is possible to be given three consecutive identical renditions of a collection of palm trees in as many moves. This is an advantage when travelling through mazes, as pictures are only displayed on as yet untouched screens, so when you get your fifth rendition of a thicket maze, you can be fairly sure of being quite lost.

A useful facility is being able to save the current state of the game on tape when you realise that it's three o'clock in the morning, or just before you do something risky like 'hit crocodile with plank'.

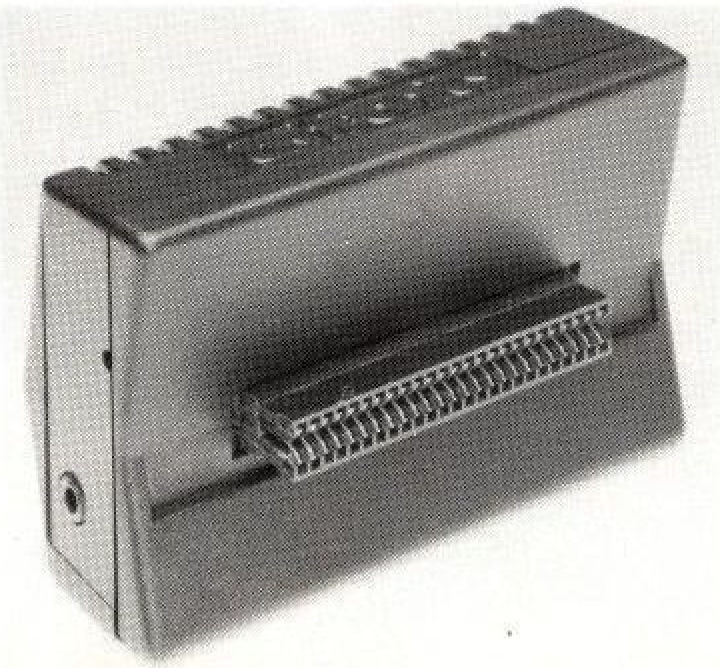
This is an interesting and challenging adventure (especially for me as I have not solved it yet), with the graphics livening things up for players who are not keen on text-only adventures.

AUTHOR	INTERCEPTOR SOFTWARE
FORMAT	CASSETTE
PRICE	£6.00
GRAPHICS	*****
PLAYABILITY	***
ADDICTIVENESS	****
OVERALL	****

Amstrad Speech Synthesizer!



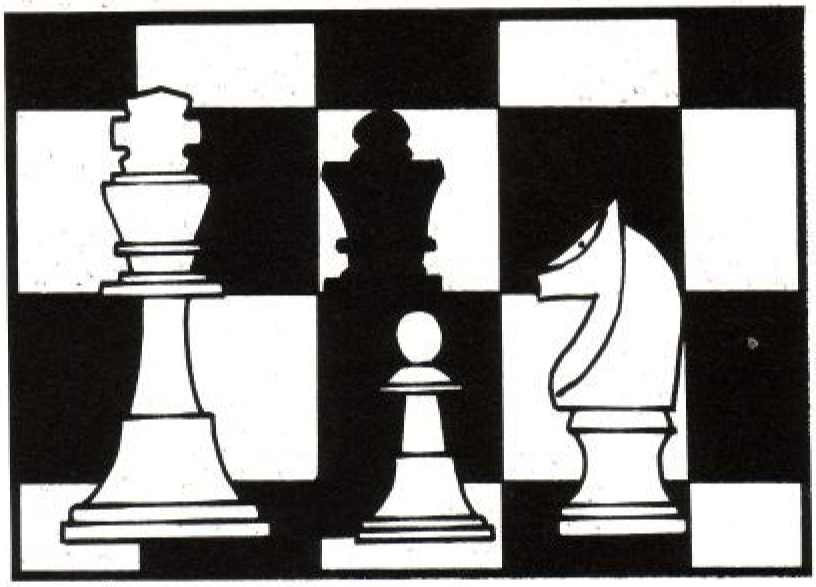
The dk'tronics Amstrad speech synthesizer and powerful stereo amplifier uses the popular SLO/256 speech chip and has an almost infinite vocabulary. It is supplied with a text to speech converter for ease of speech output creation. Everything you wish to be spoken is entered in normal English, without special control codes or characters, it is therefore extremely easy to use. The voicing of the words is completely user transparent and the computer can carry on its normal running of a program while the speech chip is talking. The speech output from SLO/256 is mono and directed to both speakers.
To utilise the Amstrad stereo output on the back of the computer, the interface has a built in stereo amplifier, this gives all sound output a totally new dimension and greatly improves the sound quality and volume over the computer's internal speaker.
Although there are only 26 letters in the alphabet, letters have a totally different sound when used in different words. For example, The "a" in Hay is much longer and softer that in Hat. When you speak you automatically make adjustments because you know just how a word should sound. Not quite so easy with a computer. After looking at other speech synthesizers we decided that it was essential that the dk'tronics Amstrad Speech would offer a simple system that would enable the user to produce realistic speech that was instantly recognisable.



The solution to the problem was extremely complicated, it required hours of programming to enable the computer to look at the individual letters that make up each word and compare their relative position to each other before deciding on the appropriate sound.
I am delighted that we have now perfected what I consider to be the best Speech Synthesizer on the market, one which has achieved my aim, within the limitations of the allophones, of producing realistic speech.
At **only £39.95** the dk'tronics Amstrad Speech Synthesizer represents remarkable value for money.
Available from department stores and good computer shops everywhere or direct from
dk'tronics, Saffron Walden, Essex CB11 3AQ. 2CPC
Tel. (0799) 26350. Add £1.25 post and packing.

Chess Master

Stephen Greenwood.



My life has changed...and I hold the CPC464 responsible! My working day now starts with loading the Master Chess program, a birthday present from my wife. It was just what I wanted, which is hardly surprising; it is, after all, what I asked for. What happened to sentiment? Well birthday surprises are fine, but this time I was desperate to get this particular 'prize'.

Let me explain. I've had this computer for three months now, and, perhaps against the trend, it's the first one I have ever owned, although I am hardly a stranger to the world of ROMs and RAMs having trained several years ago on an ICL mainframe and a Digital mini. In fact you might say I have long since been 'bitten' by the 'bug'. I have not 'entered the market' before for two reasons. Firstly I have been waiting for the right product and secondly money has been a bit tight, you see, what with my line of work being competitive, and me only starting out, as it were. Well, freelance writers are ten-a-penny aren't they? And a penny isn't worth much split ten ways.

As I was saying, it's been three months now and not one byte of commercial software has passed the lips of my datacorder. Not that I wasn't hungry for some. In fact I could swallow them all whole, from Happy Writing to the Word Processor, even though I have not yet got a printer, but I will do, believe me. Then to get rid of my outmoded sausage machine of an electric typewriter! So, why is it that I've bought not a sausage of software so far? You guessed it. I'm skint. Instead I have spent my three months working my way through the User Guide (not without its faults), trying to get to know the BASIC and entering any and all listings I can lay my hands on, debugging where necessary and improving where I see fit, generally enjoying myself and all-in-all trying the CPC464 for size. I have to report that it's a pretty snug fit although I hope to be able to buy new colour seperates in the future.

What about the chess? Well, some people like music while they work, some demand absolute silence, others cannot do a tap unless all around is chaos. I have found a new companion at my desk as I scribble, with music of one sort or another quietly filling the room like an acoustic version of wallpaper. Yes, I play chess while I write. Level 9 is too hot for me. Level 7 suits my purpose, allowing me sufficient time between moves to get some work done while providing an opponent who is constantly stretching my capabilities without making demands on my hospitality. Nor does my adversary intimidate me into moving before I am ready, although I can make him move any time I choose; a bit unfair really, but I'm not complaining.

All this has helped me to rediscover the game, which I have hardly played in the last few years for want of a regular partner. Really this piece of software is ten opponents, I can switch about at any time, even during games. But that prompts a complaint.

As I tackled the Master, fresh out of its wrapper, at level 9, I grew impatient and entered M to force a move, then modified the level to 0 for comparison. I 'checked' the black king, but the master ignored this and made its own move. I modified the board to its previous state eventually after the software had insisted on replacing the black king and a bishop with two white pawns! I was beginning to think that I had got a new adventure game called Roland in Wonderland under the wrong cover. Nothing has gone wrong since, but can you tell me what happened? Was it something I said?

The beautiful thing about computer chess is its accessibility. The board sits on my monitor screen all day painstakingly evaluating its available moves and I can look up at those awkward times when ideas fail, to study the position, nor is there any danger of disrupting the board while reaching for a pencil. No, the danger to me is that my writing will become checked with chess terminology and lose its market value. En passant, were this to happen I may have to pawn my CPC 464.

FLY DARK STAR



NOW AVAILABLE FOR THE AMSTRAD CPC 64 £7.95

Mail Order enquiries, 2 Ashton Way,
East Herrington, Sunderland SR3 3RX.

Trade enquiries:– Tel: 061-205 6003.

Design Design

IN THE DARK

VIRGIN BOOKS Virgin Books, 61-63 Portobello Road, W11

Virgin Computer Games Books

MORE Fun FOR YOUR MONEY

GAMES FOR YOUR AMSTRAD STEVE WEBB **£2.95**

More than 20 challenging and original programs guaranteed to provide hours of entertainment – Protector, Gold Bullion Raid, Las Vegas, Demon Driver and many more specially written for the series.

AMSTRAD MACHINE CODE

STEVE WEBB **£4.95**

To take full advantage of your Amstrad CPC 464 you will need to learn machine code – it uses less memory and runs up to 100 times faster than ordinary BASIC. This book will enable you to grasp the fundamentals quickly and then goes on to demonstrate in detail how to write a simple Space Invaders game.

Also in the series
USEFUL UTILITIES FOR YOUR AMSTRAD
by Steve Webb, £2.95

FEATURE

What is Forth?

Of all the many languages in the computer world, Forth is probably the most difficult to describe to people who have not actually used it. BASIC is easy and popular, Pascal is powerful, "C' is versatile, APL is compact, Logo is child's play (and fun!), but Forth is, well, just bizarre. And yet it has a huge band of dedicated followers. After BASIC and Pascal, it is probably the most popular language used on microcomputers. In attempting to describe Forth to the uninitiated, we'll also discover why it is as popular as it is.

Forth started life as a compact, powerful and (relatively) easy to use language for controlling giant telescopes. In its early days, during the 1970's, it spread by word of mouth among the astronomical community and eventually leaked out into the wider world of home micro users. What they found in it that attracted them so much was a compact language that took up little space in the limited computer memories of the time. Although Forth occupies realtively little memory, it offers the advantages of traditional "high-level" languages in requiring no assembly language programming (and no deep understanding of the inner workings of the computer) and yet it runs with the speed of an assembly language program. Indeed, programs written in Forth are often as compact as assembly language programs, run virtually as fast but can be written "at the keyboard" just like BASIC.

The biggest barrier to the wider understanding and acceptance of Forth comes from two things: one, it uses an arcane and off-putting vocabulary; two, it uses reverse Polish or "postfix" logic in its program construction.

We'll start by illustrating both of these "shortcomings" in a mini-program in BASIC and Forth. First the BASIC version:

```
10 LET X = 7
20 PRINT X
30 PRINT CHR$(X)
```

All BASIC programmers will recognise this as a program that first assigns the value 7 to variable X. It then print the value of X on the computer screen and then "prints" the character, which in this case is the "bell" or beep sound. An exactly equivalent program in Forth would look like this:

```
VARIABLE BEEP
7 BEEP !
BEEP @.
BEEP @EMIT
```

The first thing that will strike most programmers is, how do you read it? What are all those "@", "." and "!" signs, and what do they mean? In BASIC, the PRINT statement is fairly self explanatory as long as you interpret the meaning of "print" somewhat loosely, and PRINT CHR$(something) will also be meaningful as long as you know that CHR$( ) is a function that returns the character value of a numeric argument (variable). One of the problems of Forth is that many of the "words" equivalent to BASIC's PRINT and LET are unfamiliar signs that do not have an obvious way of being read. BASIC's PRINT is simply . in Forth, while LET is ! (pronounced "store"). These "shorthand" words are not fixed and immutable, as they are in BASIC (or Pascal or most other languages). The programmer can easily redefine the name of words in Forth, and create new ones too. Suppose for example, that you dont happen to like the word PRINT for the instruction that causes characters to appear on the screen, and would prefer it to be called SLURP. What can you do about it? Nothing. Unless, that is, you're willing to rerwite the entire BASIC interpreter. In Forth, however, if you don't like "." as the instruction to print, you can create a new one as easily as this:

```
: SLURP . ;
```

Thereafter, at any point in the program where you would have had to use "." to print a character, you could use SLURP. If you were more conventionaly minded, and would rather use PRINT than . you could redefine it thus:

```
: PRINT . ;
```

Any line in forth starting with a ":" and ending with a ";" is called a definition. The first word (or symbol) after the colon is then defined in terms of whatever follows it. The definition is terminated with the semicolon ";".

Using Forth definitions in this way allows existing "words" such as "." to be re-defined, but can just as easily be used to add new words to the Forth vocabulary. On the computer system used to write this article, the screen can be cleared by "printing" ASCII 12. In BASIC this would be done by using the line PRINT CHR$(12), although most versions of BASIC have a command CLS that does the same thing. Forth's equivalent of PRINT CHR$(12) is 12 EMIT. To add the word CLS to Forth so that it has the same effect as it does in BASIC is as simple as this:

```
: CLS 12 EMIT ;
```

Once that has been entered, all You have to do to clear the screen is type CLS. The word has now been added to Forth's vocabulary, to use Forth jargon.

The Forth, the computer user buys, consists of a small kernal of machine code routines running in a loop. This loop constantly checks input from the keyboard and tries to interpret anything typed in. The kernal recognises a limited set of Forth "words" and has an in-built ability to redefne new words in terms of old words (as we saw in the examples above). This kernal provides the programmer with all the facilities needed to write basic programs and to define new, more complex, operation in terms of existing, pre-defined operations. The language is one of a small class of languages known as "threaded interpretive languages" and is fundamentally different from the ususal interpreters and compilers of BASIC, Pascal, C and so on.

As you probably know, interpreters have a built-in editor that allows source code (original programs) to be entered at the keyboard. As soon as the RUN (or equivalent) command has been issued, each line of code is translated (interpreted) into machine code for immediate execution. With a compiled language, such as Pascal, the source code is entered using as word processor or other editor to create the source code and then the language compiler converts the whole program to "object code" or a machine language

lent of the source code that the computer can execute. Forth is neither of these. Instead, it is simultaneously both an interpreter and a compiler, and this is a further dimension to the confusion that exists concerning Forth. When a program is typed in at the keyboard, each line is interpreted much as it would be with BASIC. However, when definitions are added to Forth, using the : instruction string ; sequence, the new definition is compiled and added to the Forth kernal. An applications program will be built from existing kernal definitions and pre-compiled new definitions, so when the new program is actually executed, its lines will be interpreted using precomiled definitions.

This is what gives Forth its speed compared with languages such as BASIC, and it is the small size of the original Forth kernal that gives it its compactness.

Reverse Polish Notation

One of the problems that new users of Forth encounter is its use of Reverse Polish notation. We are acustomed to performing arithmetic in so-called "infix" order, so 2 times three plus two will be written 2x3+2. In BASIC this would be written 2*3+2, as in: LET X = 2*3+2. Forth works differently. First you enter the first argument (first number), then the next argument, then the operator ("times" in this case). Then any further operator is enterd, followed by any further operator. 2*3+2 in Forth would be entered:

```
2 3 * 2 + <CR>
```

Forth makes extensive use of what is known as a push-down stack. Values are entered from the keyboard and entered by Forth on the top of a stack of numbers. The last number on the stack ushes down all previous numbers, just like the plates supported by a spring in some transport cafes. When a number is retrieved from the stack, it is always the last number eneterd that comes out first. In the example above, first the number two is pushed onto the stack. Then the number 3 is pushed on on top of the 2. The * (multiplication) operator then pops these two numbers from the stack, multiplies them and pushed the result onto the stack again. Then a new 2 is pushed on top of the previous stack item (now 6). The + operator then pops the top two items from the stack (6 and 2) and adds them. The result (8) is then pushed back on to the top of the stack. If you remember that "." in Forth means "print", the following line will produce the following result:

```
2 3 * 2 + . <CR>
8 ok
```

The meaning of the "ok" is that Forth has executed the line and that no errors have occured. In Forth, arguments are passed almost always via the stack, and so variables and constants are used much less frequently than they are in BASIC and other languages. In BASIC, all possible variabels exists implicitly, so you can write, in the middle of a program, say, `LET TIME$=13:4000` there is no need to specially create a variable called TIME$. Forth has much less need of variables, and when one is required, as in `BEEP` above, it must be specifically created by declaring it, as we did in `VARIABLE BEEP`. Values can then be stored in it by using the "store" word "!" or retrieved using the "fetch" word @ (pronounced "fetch").

Using Forth on the CPC464

All you have to do to run Forth on a cassette based CPC464 is to load the cassette and type `RUN "FORTH"`. The "ok" primpt shows that Forth is ready to run. You can test that everythig is working properly by entering

```
:TEST100 0DO."HELLO"LOOP;[CR]
ok
TEST[CR]
```

When you hit **[ENTER]**, the word HELLO will be printed out on the screen.The line of Forth program is an example of a `DO LOOP`. The first entry, 100 is pushed onto the stack as the limit of the loop. the second number, 0, is pushed on as the starting point. "." means print the following sequence as terminated by the next double quote sign and the final `LOOP` is the instruction that makes the program go back to the `DO` part until the loop has been executed the right number of times. The equivalent in BASIC is:

```
FOR X = 1 TO 100
PRINT "HELLO"
NEXT X
```

As these few simple examples will have shown, programs can be written in Forth almost as simply as they can in BASIC. Forth's great advantage is that the programs run much faster, and they also take up less space in memory. The power and speed of Forth is the reason why so many arcade games from people such as Atari are witten in Forth. It's a unique language that gives the speed of assembler but is much easier to learn.

Although not quite as easy to program in as BASIC, it is far more flexible. For the serious programmer, interested in modular "structured" programming and self-documenting source code, Forth has a lot to offer. Names of functions can be freely chosen and new functions built up from previously tested, simpler functions. Consequently, a complete, ready-to-run program could look as simple as this:

```
: POKER
GREETING
BEGIN
PLAY-GAME
GOODBYE-MESSAGE
AGAIN
;
```

Each part of this program, namely GREETING, PLAY-GAME and GOODBY-MESSAGE will consist of a series of previously written and tested definitions. All that you have to do to play the poker game is to type POKER.

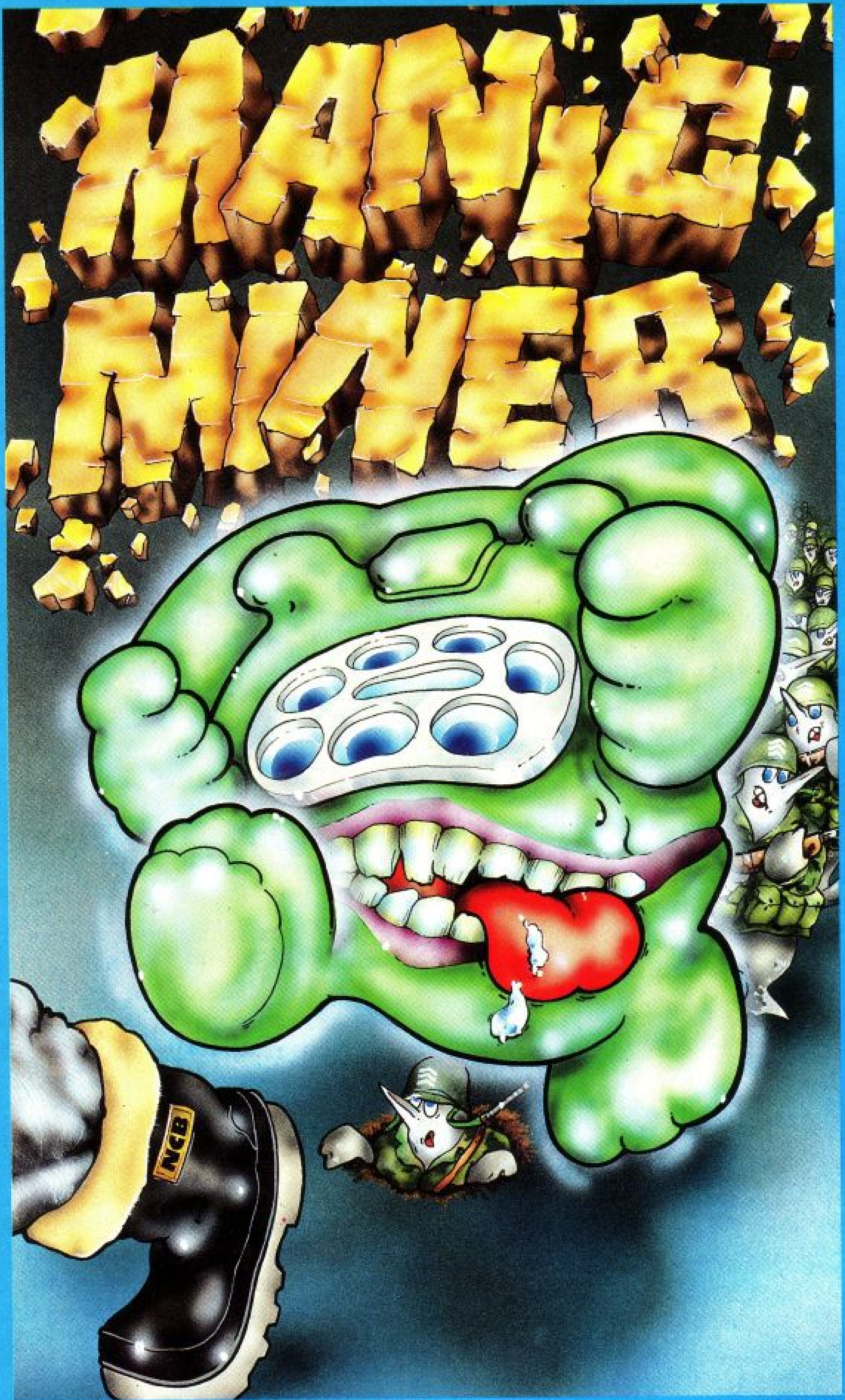
On the debit side as far as readability goes is the fact that Forth encourages the programmer (but does not force him) to pass variable values via the stack instead of via named variables. Just pushing values onto the stack and popping them off (without using named variables) can make Forth programs run like greased lightning but makes them very difficult to read.

Finally, here is a short random number generator taken from the publications Forth Dimensions. This one has been tested on Abersoft's Forth, and it runs just fine.

Forth Definitions Decimal

```
0 VARIABLE SEED
: (RAND) SEED @ 259 * 3 + 32767 AND DUP SEED ! ;
: RANDOM (RAND) 32767 */ . ;
```

Type 100 RANDOM and a number between 0 and 100 will be printed on the screen at random. Don't forget to type RETURN at the end of each line in the program above.









Miner Willy, while prospecting down Surbiton way stumbles upon an ancient, long forgotten mineshaft. On further exploration, he finds evidence of a lost civilisation far superior to our own, which used automatons to dig deep into the Earth's core to supply the essential raw materials for their advanced industry. After centuries of peace and prosperity, the civilisation was torn apart by war, and lapsed into a long dark age, abandoning their industry and machines. Nobody, however, thought to tell the mine robots to stop working, and through countless aeons they had steadily accumulated a huge stockpile of valuable metals and minerals, and Miner Willy realises that he now has the opportunity to make his fortune by finding the underground store. In order to move to the next chamber, you must collect all the flashing keys in the room while avoiding nasties like Poisonous Pansies and Spiders and Slime and worst of all, Manic Mining Robots. When you have all the keys, you can enter the portal which will now be flashing. The game ends when you have been 'got' or fallen heavily three times.

MANIC MINER is available on Spectrum, Amstrad, BBC, Dragon, CBM 64, Oric/Atmos, MSX.

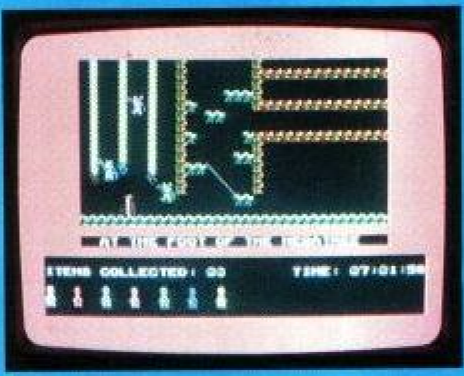
The above screens are from the BBC version.

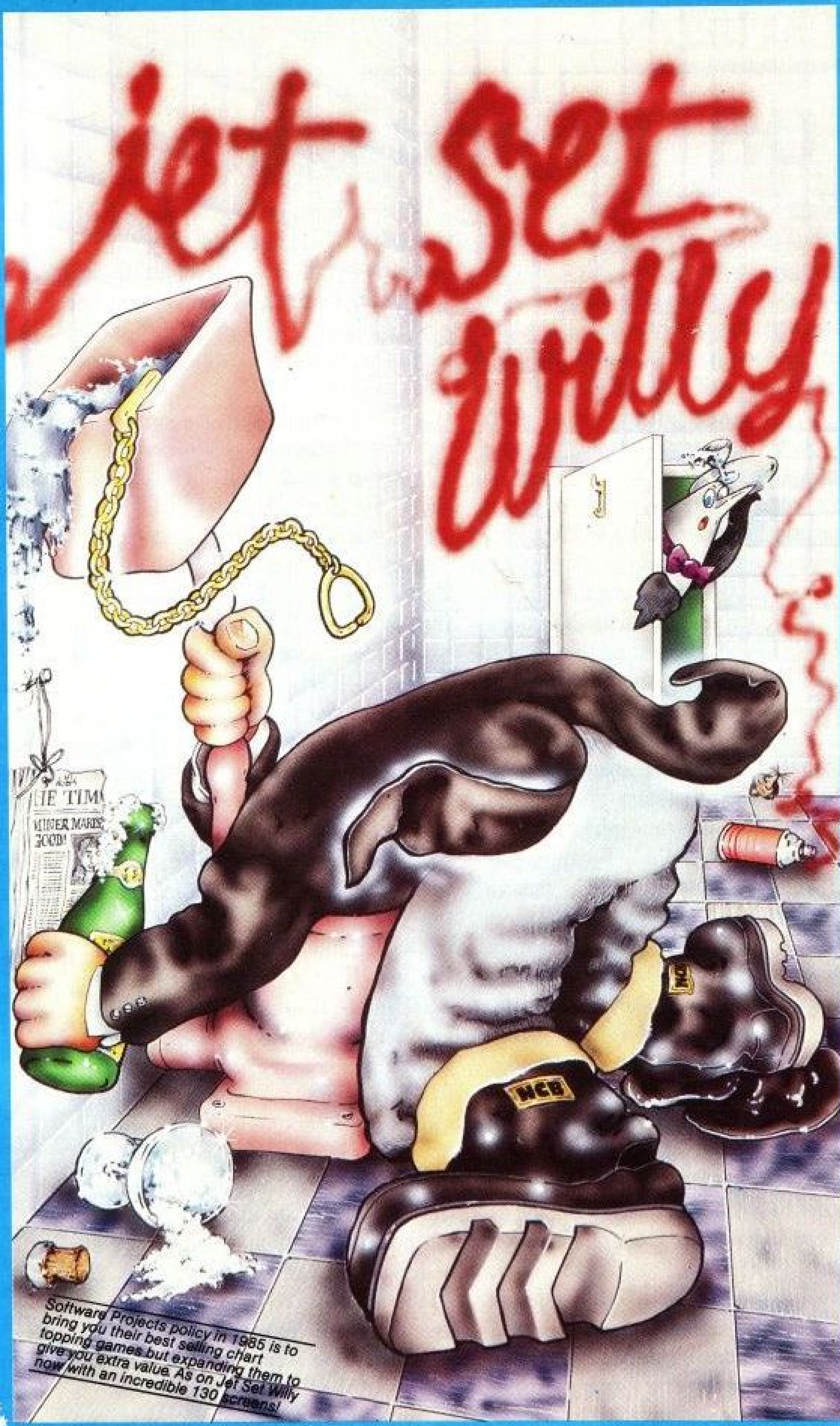
SOFTWARE PROJECTS

Software Projects Limited, Bearbrand Complex, Allerton Road, Woolton, Liverpool L25 7SF. Telephone: 051-428 9393 (4 lines). Telex: 627520.









Miner Willy, intrepid explorer and nouveau-riche socialite, has been reaping the benefits of his fortunate discovery in Surbiton. He has a yacht, a cliff-top mansion, an Italian housekeeper and a French cook, and hundreds of new found friends who really know how to enjoy themselves at a party. His housekeeper, Maria, however, takes a very dim view of all his revellery, and finally after a particularly boisterous thrash she puts her foot down. When the last of the louts disappears down the drive in his Aston Martin, all Willy can think about is crashing out in his four-poster. But Maria won't let him into his room until all the discarded glasses and bottles have been cleared away. Can you help Willy out of his dilemma? He hasn't explored his mansion properly yet (it is a large place and he has been very busy) and there are some very strange things going on in the further recesses of the house (I wonder what the last owner was doing in his laboratory the night he disappeared). You should manage O.K. though you will probably find some loonies have been up on the roof and I would check down the road and on the beach if I was you. Good luck and don't worry, all you can lose in this game is sleep.

This is a Computer and Video Games Hall of Fame Game.

JET SET WILLY is available on Spectrum, Amstrad, CBM, MSX.

Send a large stamped addressed envelope for more detailed information on our new releases, plus a free large poster.

All sales enquiries to:
Colin Stokes
(Sales and Marketing)

For Mail Order only:
Software Projects, P.O. Box 12, L25 7AF

Selected titles available from:–
John Menzies, HMV, Boots, W. H. Smith, Woolworth, Spectrum and other large department stores and all good major software retailers.

Please send me a copy of — **Please tick were applicable**

JET SET WILLY · AMSTRAD £8.95 ☐
MANIC MINER · AMSTRAD £8.95 ☐
JET SET WILLY · MSX £7.95 ☐
MANIC MINER · MSX £7.95 ☐

I enclose cheque/PO for ....

(Please add £1.00 for orders outside UK)

Access

Access Card No. ....

Name ....

Address..

....

For Mail Order only:
Software Projects, P.O. Box 12, L25 7AF

FORTH For Beginners

Sooner or later, most programmers feel the need to learn another computer language basides BASIC. This may be for reasons of speed, effeciency, or just plain curiosity. The choice on a home micro is likely to be between FORTH, PASCAL and ASSEMBLER.

Many programmers attempt assembler - and nearly as many give up in despair. Learning to program in Pascal or Forth is perhaps not so daunting, but considerable effort will still be needed to get to grips with either. In this series of articles I shall be looking at my favourite language FORTH. In the space available I shall only be able to cover the fundamental basics of the language, but I hope to discuss some of the more advanced features in brief in the last article.

The best way to learn Forth is by 'hands-on' experience. Unlike Pascal, and similar to BASIC, Forth is an interpretative language. This means that commands (known as WORDS in Forth) can be entered and the results displayed immediately. This is the same as the 'immediate' mode in Basic. Forth is not compiled into machine code it is generally referred to as a 'threaded interpretative' language, however a form of compilation does take place, as will be seen shortly.

Before beginning to program in Forth it would be a good idea to have a look at its overall structure. The Forth as used on the CPC464 is a Fig - Forth (Forth interest Group - Forth) implementation. You may also come across the Forth-79 Standard, but in general the difference between the Forth variants is small - much less so than the numerous Basic around.

Perhaps the best thing to do after loading the Forth program into your CPC464 is to VLIST - and remember to press the Enter Key. The screen will quickly fill up with a list of Forth words. Many of the words in the first six lines will be familiar to you as CPC464 BASIC commands. These are not standard Forth words but are 'extension' words that have been added to take full advantage of the CPC464's special features - sound and graphics, for example. In general, all versions of Forth will have extensions provided to suit the micro in question.

Of the words displayed, some will produce immediate effects when entered -`CLS`, `VLIST` and `CLG` for instance. Other words require values such as `BORDER`, `PEN`, `PAPER`. Some words fall into either category. These are the Defining words or control structures such as `IF`, `THEN`, `REPEAT...UNTIL` `DO...LOOP`.

The whole collection of words is known as the FORTH DICTIONARY. This dictionary is further subdivided into VOCABULARIES. The list of words presently on the screen are those in the FORTH VOCABULARY. This amounts to just under 300 words - and shows how useful an 80 column screen is ! Nestling amongst these words are two further vocabulary words EDITOR and ASSEMBLER. Type EDITOR and then VLIST and you will find that an extra 28 words have been included at the beginning of the list. These 28 words make up the EDITOR VOCABULARY. Now try entering ASSEMBLER and VLIST. This time another 160 words of the assembler vocabulary are displayed. Not that the EDITOR words have now gone. Any number of vocabularies can be created, but only two can be used at any one time - and one of these must always be the Forth vocabulary. Some simple mathematics will show that there are nearly 500 words available on the AMSOFT FORTH. Compare this to the average of 120 for most basics and you can appreciate why I can only skim the surface of the Forth language. The use of vocabularies in Forth is an important, and often over looked one. Figure 1 shows the layout of a Forth Dictionary.

These new words can then also be used to make up yet more new words. Each word will progressively become more powerful. The complete program will enventually be represented by one word only. The ability to define new words is similar to creating new Basic words on the CPC464 with the RSX facility. To use the RSX however you need to be a reasonably competent machine code programmer. No such skills are required in Forth, although if you can program in machine code full advantage can be taken of this with Forth.

The most common method of creating new words is the DO: word. Let's imagine that we wanted to make the existing words more 'user friendly', useful for a complete novice to programming. We all know that CLS stands for 'clear the screen', but is it so obvious to a novice ? It's doubtful. So in Forth you could say:

```
: CLEAR-THE-SCREEN CLS ;
```

The colon tells the Forth interpreter that the next word that follows is a new word. A space in Forth acts as a delimiter between words, so it is necessary to join up a new word with a dash, like in the above example, or a dot, or anything you like. If you had left out the dashes in the above example Forth would take CLEAR as the new word. As THE is not a standard Forth word this would create an error message-

```
THE? MSG #0
```

which it telling you that Forth has searched through all the words in the current vocabulary(s) and has not found THE.

Try typing in the above example and Forth should reply with 'OK'. This is

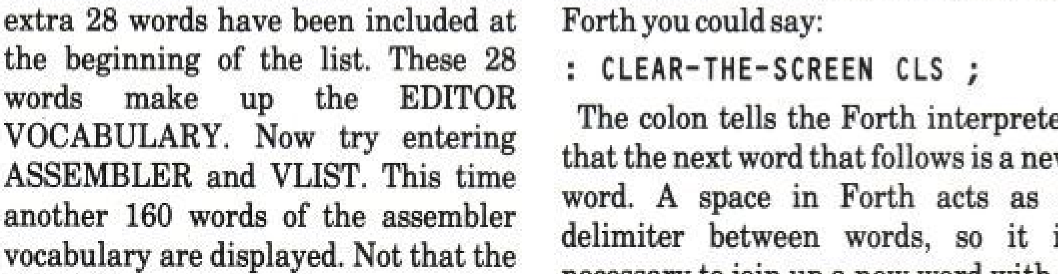


Figure 1

The arrows show how the individual words are linked.

The main idea of Forth is to make up new words from the existing words.

the standard Forth word to let you know that, in this case, the new word CLEAR-THE-SCREEN has been successfully compiled into the

dictionary.

The semi-colon marks the end of the new word. Now try typing in CLEAR-THE-SCREEN and the screen should indeed clear. Try changing VLIST to:

```
PLEASE.DISPLAY.A.LIST.OF.
THE.FORTH.WORDSVLIST ;
```

Now enter

```
PLEASE.DISPLAY.A.LIST.
OF.THE.FORTH.WORDS
```

and the list of words will be displayed. I know what your're thinking.... what a lot of typing. I agree, but remember that in the near future computers will probably use voice input recognition instead of a keyboard. You could literally talk to Forth in plain English (or any foreign language for that matter). using whatever words you like, Forth could be the most user freindly of all programming languages.

The length of a new word can be up to 127 characters. The first 31 characters are all significant as is the total number of characters in the word. In the above example the characters are significant up to the '0' in FORTH. The length of the word is also stored, which is 40. If you added:

```
.TO.THE.PRINTER
```

to this word to create another new word Forth could differentiate between the two, as the length of this word would be 55. Any ASCII character can be used in a new word, apart from a space. The following words would all be legal, and different, in Forth:

```
AMSOFT
CPC464
99trombones
*++)(Ø!!
DRAW;A;SQUARE
347.ØØ79
1ØØK
not.the,same
not,the.same
```

As you can see, just about anything goes in Forth! Previously defined words can be embedded and upper and lower case characters are treated as being different - which of course they are - but not in the majority of BASICS.

Previously used words can also be used again although you will get the MSG # 4 (ISN'T UNIQUE) warning you of this. The range of new words that you can create is therefore huge. Be sensible in your choice of words though. For example, when you first load in the Forth program you could make up the following new word:

```
: VLIST CLS ;
```

Forth will issue the MSG XXX 4 warning telling you that VLIST already exists as a word. VLIST has been compiled into the dictionary though as you will find out if you enter VLIST - when the screen will clear! what has happened to the original VLIST though ? Well it's still there in the dictionary, but as Forth searches through its words from the most recently defined it will execute the VLIST that we have just created above. The original VLIST is now redundant -it will never be reached! Fortunately the word FORGET can be used which deletes the word specified, and all the words above it in the dictionary. Enter:

```
FORGET VLIST
```

and the VLIST which clears the screen is deleted from the dictionary and the original VLIST is restore. The words that come with your Forth package are 'protected' and cannot be 'forgotten', although it is possible to lower the 'fence' which marks the position below which words are protected. It's also possible to place this fence after the new words you have created, thus making them 'protected'. If you save this new version of Forth to tape or disk, then these new words will become an integral part of your Forth language. It is easy to see that you could create various Forth packages to cater for various applications e.g. Mathematics, Word Processing, Graphics, Financial etc.

If you have tried out these new words in the article, then you should see them included on the VLIST. The last word created is always the first word on the list. One problem will arise which is rearely mentioned in Forth books. We all make typing mistakes - forgetting to leave a space is the most common with Forth, or putting a space in a word name when there should not be one. Try this:

```
: ERROR! CLS V LIST ;
```

Forth will reply with V? MSG # Ø We have put a space between V and LIST when there should be none. The intrepreter tries to find V in the vocabulary. It is not a standard Forth word so the address of the word V doesn't exist. Well no harm has been done so try again with:

```
: ERROR! CLS VLIST ;
```

Although not a very good choice of word ERROR! will now be compiled into the dictionary. 'OK'. Now VLIST out the words (you could use ERROR!). You should see that the word ERROR! appears twice. The first one is the correct one, the next one is the incorrect one. Even though it was incorrect it has been compiled into the dictionary - up to the point at which the mistake was found. However, a SMUDGE bit is set at the beginning of this word. Anywords that have the smudge bit set, are ignored during the searches, and are not classed as part of the dictionary. They are still there though, and as such are using up valuable memory space. Fortunately Forth is a very compact language. Our incorrect word is actually using up 13 bytes of memory elthough you may enter a long word definition incorrectly which may use up hundreds of bytes. Unfortunately, if you enter an incorrect word, you cannot FORGET it. Why? Well, quite simply, when Forth searches for the word that you want to FORGET it cannot find it as the smudge bit is set!

Fortunately, the word SMUDGE can be used to 'toggle' the smudge bit in the last new word created. If you enter a word incorrectly then SMUDGE it and FORGET it. For example, after entering ERROR! incorrectly you should type:

```
SMUDGE FORGET ERROR!
```

ERROR! will then be deleted from the dictionary. Not that SMUDGE can only be used on the last word created, and smudging a word that was compiled successfully will have the opposite effect and make it redundant, as it will now be ignored during any search.

In the next article we will get to grips with two of the most important aspects of Forth - the STACK, and the much discussed REVERSE POLISH notation.

EASIDATA — The ultimate database program for your CPC464. Up to 32 items per record. Up to 1,000 records. Full search facilities, including wild card. Records automatically inserted in correct order. Built-in report generator allows simple word processing with mail merge and label printing. Outstanding value at **£19.99.**

EASIVAT — Makes VAT book-keeping as simple as pressing a button. Up to 10 Income and 20 Expenditure headings. Up to 10 different types of transaction (e.g. cash, cheque, credit card, etc.). 100 entries per month. Quarterly VAT report. Excellent range of monthly and annual summaries. Comes complete with manual and demonstration file. An absolute must for the self-employed or small trader. **£39.99.**

DEAN DATABASE / MAILING LIST — Incredibly easy to use and flexible electronic card index. Up to 50 items per record. Up to 360 records. Fast sort and search facilities. Simple label printing routine with multiple copy facility. Large files may be split into smaller files. Excellent value at only **£12.99.**

MUSIC COMPOSER — Superb fun for all the family. Suitable for the beginner or experienced musician. Clear graphics displays the music as it is composed. Full editing facilities. Compositions may be joined together and/or saved on tape. **£9.99.**

All prices include VAT, postage and manual. To order simply send cheque or P.O. to:

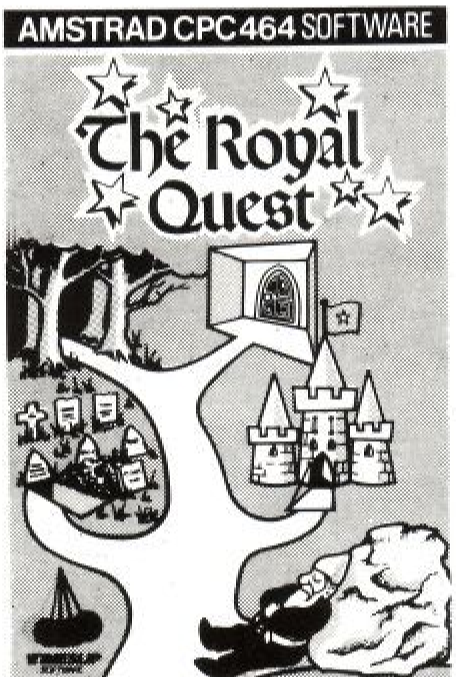
DEAN SOFTWARE LTD.

P.O. BOX 1, THE MEWS, MITCHELDEAN, GLOS. GL17 0HH

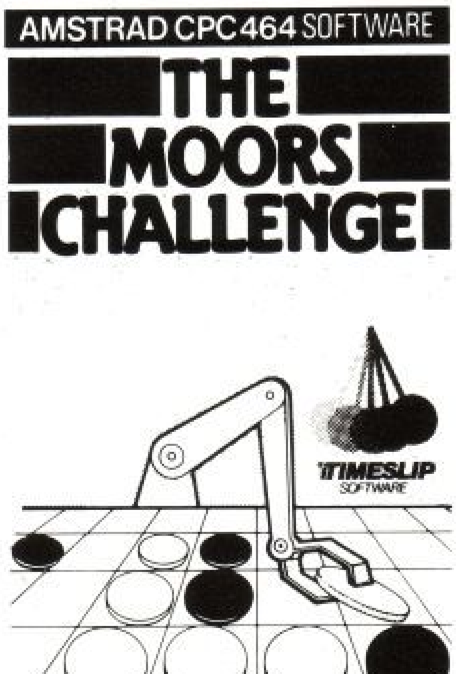
(The above versions are cassette based, disc versions available soon — details on request)



ADDICTIVE....



PERPLEXING....



CHALLENGING.

WHAT MORE DO YOU WANT?

All programs £6·95 inc p&p from

TIMESLIP
SOFTWARE
STONEYBURN WORKSHOPS.
THE OLD PRIMARY SCHOOL. MAIN STREET.
STONEYBURN. WEST LOTHIAN. SCOTLAND EH47 8AP.

Dealers Enquiries Welcome

....this month invites the MASTERFILE 464 author to describe his database program

MASTERFILE464

John A Campbell

MASTERFILE 464 is the definitive RAM database for the CPC464 computer, and is a must for the serious user. Although the program is very new, it is in effect a further development of its namesake which runs on the ZX Spectrum, and which has undergone almost continuous refinement since it first appeared over two years ago. The result is a database system considerably in advance of even the Spectrum MASTERFILE, and one which promises to be of central importance for further generations of microcomputers.

The philosophy behind MASTERFILE is one borrowed from some big-system database systems, and in that the way which data is stored should be independent of the way in which it is presented. Further, one should be able to present data in a variety of formats, ranging from a simple summary index to a full dossier. Thus MASTERFILE looks after the file handling and data access - the user then decides how he/she wants to display the data.

As should any respectable database, MASTERFILE offers unlimited searches of the file, on the basis of comparing data with character or numeric arguments, or on the basis of data being present or absent, or by scanning for strings. The program goes even further, offering to treat upper/lower case as equivalent or otherwise, when making its comparisons.

In the family

With MASTERFILE 464, we believe we have broken fresh ground by providing the first RELATIONAL database for the home micro. With other small-system filing systems - even Spectrum MASTERFILE - a file is just a list of similar but unrelated records. This is fine for many compartively simple filing uses, but a new dimension of computing power is opened up by allowing records to RELATE to each other, so that access to a CHILD record automatically gives access to the data contained elsewhere in a matching PARENT record.

All child records which link to the same parent will share the parent data, and any update to the parent is automatically reflected when viewing the related child records. Another way of looking at this is that MASTERFILE 464 allows two files to co-exist in one, and component records of one sub-file share data with linked records of the other sub-file.

Just to give an example of use, a doctor could keep a file of patient data together with prescriptions given. Clearly, one only wants to store a patient's fixed details, such as address and medical number, once. But for each prescription the doctor would create a fresh record, relating to the patient details. With such a file, one could select a patient and display his full particulars, followed by a chronological list of prescriptions. Another display format of the same file could show just patient details, or address labels for a mail shot, and yet another display could provide an analysis of the prescriptions, independent of patient data.

The SEARCH functions of MASTERFILE 464 include instant isolation of child, parent, orphan, child-less parents...sounds like the DHSS should be interested.

MASTERFILE 464 is totally machine-coded for compactness and speed, and is fully menu-driven. The user is presented with questionnaires on the screen when designing his display formats, and quite sohpisticated formats are quickly developed. The formats themselves are stored as part of the file, so creating them is not an everyday task.

All data is stored as ASCII characters, and any field can be displayed word-processed within its own window space. Field sizes are nominally limited to 240 characters, ample room for a complete address. Indeed, special logic assists the printing of address labels, forcing line-breaks where desired. For single-line data, one can right-justify it. Or, if it looks numeric, one can opt for numeric editing, inserting currency symbol, thousands' commas, and aligning on the decimal point. To add a professional touch, displays can be ruled or boxed in any manner.

To browse through a file, you can go forward 1 page, forward or backward 1-9 records, go back to the first or go straight to the last.

Numeric data can be column-totalled.

The entire file can be sorted and re-sorted on demand -ascending or descending, alphabetically or numerically - all at machine-code speed. The file can be sorted by any field, or multiple fields.

There is even a facility to search on embedded keys, such as surnames (Smith) within fields (Mr H W Smith) - no need to keep names back-to-front just for sorting purposes.

Any record can be erased (but it asks 'Are you sure?'), copied, updated. The update options include edit-mode alteration of existing fields, and you can also insert new fields, and erase old ones.

And MASTERFILE is only 10K in length, allowing over 30K for storage.

CHILTERN COMPUTER CENTRE

AMSTRAD — HAVING PROBLEMS GETTING SOFTWARE? — AMSTRAD

LOOK NO FURTHER!

Games

SPANNERMAN	8.95
ROLAND ON THE ROPES	8.95
CUBIT	8.95
QUACKAJACK	8.95
ROLAND IN THE CAVES	8.95
AMERICAN FOOTBALL	8.95
PUNCHY	8.95
AMSGOLF	8.95
ROLAND GOES DIGGING	8.95
GEMS OF STRADUS	8.95
HARRIER ATTACK	8.95
DUNGEON ADVENTURE	9.95
RETURN TO EDEN	9.95
LORDS OF TIME	9.95
COLOSSAL ADVENTURE	9.95
SNOWBALL	9.95
ADVENTURE QUEST	9.95
TEST MATCH	6.95
GRAND PRIX DRIVER	7.95
ATOM SMASHER	7.95
3D-MONSTER CHASE	7.95
ALIEN BREAK IN	7.95
STEVE DAVIS SNOOKER	7.95
JOHNNY REB	6.95
BRIDGE PLAYER	7.95
GHOULS	6.95
HUNCH BACK	7.95
MANIC MINER	8.95
FOOTBALL MANAGER	7.95
FOREST AT WORLDS END	5.50
BLAGGER	7.95
FANTASIA DIAMOND	7.95
JEWELLS OF BABYLON	5.50

Educational

HAPPY WRITING	8.95
HAPPY LETTER	8.95
ANIMAL VEGETABLE MINERAL	8.95
WORLD-WISE	8.95

CARRIAGE FOR COMPUTER, PRINTER, DISC £7.00
CARRIAGE FOR JOYSTICK, LEAD, MODULATOR £1.00

Utilities

TASWORD 464	19.95
EASI-AMSWORD	8.95

Hardware

AMSTRAD CPC464 COLOUR COMPUTER	342.50
DMP-1 PRINTER	195.00
DISC DRIVE DD1-1	199.95
AMSTRAD JOYSTICK	12.95
PRINTER LEAD	9.95
POWER MODULATOR	29.95

MAIL ORDER

PLEASE MAKE CHEQUES & POSTAL ORDER PAYABLE TO:
"CHILTERN COMPUTER CENTRE"
76B DALLOW ROAD,
LUTON.
BEDS LU1 1LY
TEL (0582) 455684

WE ACCEPT VISA — OVERSEAS ENQUIRIES WELCOME

CRYSTAL THEFT

FROM WICCA-SOFT

An adventure set in the early days of the Vegan war. It is 2319, and your mission is to direct Field Captain Jonathan Markham of Earth Armed Services into an abandoned Temple to steal a communicator crystal.

How abandoned is the Temple? Can you solve sometimes complex puzzles IN REAL TIME? What use is an accurately machined heavy metallic cube? Can you disable a laser barrier? Will you recognise the final problem before it kills you? Over 90 locations will lull you into a false sense of security, test your wits and may even destroy you.

Available late November for the Amstrad CPC 464



PRIORITY ORDER FORM — Dealer Enquiries Welcome

Please rush me AS SOON AS POSSIBLE _____ copies of CRYSTAL THEFT at £7.99 each (inc p+p). I enclose a cheque/P.O. for £ _______ (NO CASH PLEASE)

PRINT IN BLOCK CAPITALS

NAME ____________________

ADDRESS ____________________

POST CODE ____________________

Post to: Dept TS, WICCA-SOFT, 107 Bollington Road, Bollington, Macclesfield, Cheshire SK10 5EL

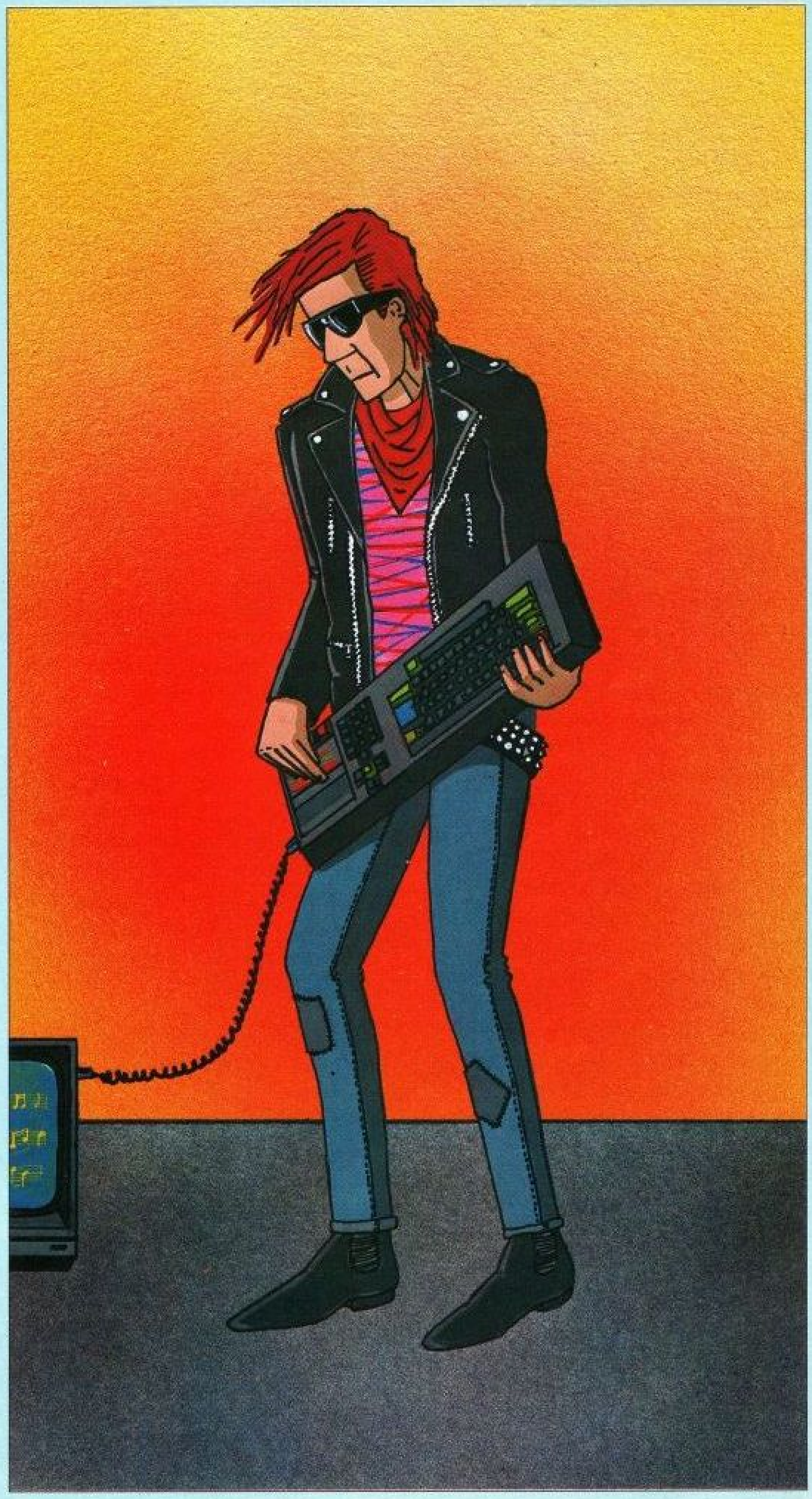
Allow 21 days for delivery

WICCA-SOFT

INTRO TO MUSIC

Part 1

JEREMY VINE



The CPC464 is many things to different people and amongst its various talents it boasts an elaborate sound generation facility that can produce anything from 'laser zap' blasts to keyboard music. In this, the first of a series of articles on creating music, I'm going to look at one of the commands responsible for creating sound from BASIC.

At this point don't get put off by the mention of 'music'. You don't have to read music to play it on your CPC464 and I will be introducing the rudimentary elements of music as we proceed. But first some background information.

Inside your Amstrad is a sound generation chip called the AY-3-8912 produced by General Instruments. It is this chip which is responsible for creating sound. But what is sound? Now that may seem a daft question but sound has many complex properties, and a basic understanding of what sound is will help you use the sound chip to its fullest capabilities.

The three most basic concepts of sound are the duration, amplitude and pitch of a sound. These three elements are at the roots of sound generation on the computer so let's first look at these in turn.

Duration

This refers to the length of a sound in time. The duration of a sound in musical terms is important as it can both determine the rhythm of a piece of music and how loud a note sounds. Now that last point may seem odd to you but a sound or noise can appear louder or softer depending on the length of the sound. A very short tone will sound softer than a longer note.

In musical notation duration plays an important part and later we will see how the duration of a note is indicated to the musician. Duration is also one of the basic elements of the first BASIC command we're going to look at, that



being the SOUND command. But before we do let's take a look at the other two main factors involved in sound production.

Amplitude

I've already touched on the subject of amplitude in the explanation of duration. Simply speaking it is how loud or soft a sound appears to the ear. On the CPC464 there are two main ways of altering the volume level of a sound. Firstly by means of the volume dial on the right hand side of the machine or secondly by means of software control. Now this may seem to you to be the end of the story. Amplitude is merely the volume level of a noise, right? Well, not quite!

Like life, things are not so straightforward. Within a given sound other amplitudes can be hidden away and these are not always apparent to the listener. If you think about the sound of a particular musical instrument, each has a distinctive sound because of the way the volume varies within the note played. I'll come back to a more detailed explanation of this when we look at the envelope commands (ENV and ENT). The final building block of sound is the pitch.

Pitch

The pitch of a note is very simply whether a sound is high or low. To put it another way, pitch refers to the frequency of a sound. As you know sound travels in waveforms and the number of times a particular waveform vibrates per second determines its frequency. This is usually measured in Hertz (Hz) which is the number of cycles per second. Our hearing as human beings is restricted to certain frequencies where we can hear sound between roughly 20 to 20,000 Hertz.

A well know illustration of this limitation is the dog whistle. Because dogs can her at a higher frequency than humans, they're able to detect the whistle even though we cannot hear it ourselves. So just because we can't hear a sound doesn't mean it's non-existent.

For the purpose of this series of articles, pitch will refer to the accepted frequencies in western music like those of a musical scale. On a piano keyboard you have a fixed sound which can be played at different pitches, high or low depending on the key struck at the keyboard. The pitches ascend from the far left of the keyboard (a low note) to the far right (a high note). These frequencies of sound are at fixed regular intervals.

The CPC464 allows the user to control the pitch of a sound very precisely and we will see how it can be manipulated by means of the ENT command later on. But let's take a break from what is possible on the computer and consider some musical notation.

Music

We all know what music is but if you've never played an instrument or read music the thought of all those funny looking characters might make you think that learning Arabic is easier! But don't give up. Learning the basics of music is not that hard and it would be a pity not to get the most from your computer. I'm going to assume that you're acquanited with the convention of naming musical notes, if not I suggest a quick look at the first few pages of a rudimentary music theory book, or if the editor doesn't cut this out, I have a certain bias to my own book which includes explanations of the ins and outs of musical notations!

To start you off, the range of possible notes is as follows:

C C#/D♭ D D#/E♭ E/F F#/G♭
G G#/A♭ A A#/B♭ B

In western music there exists a range of twelve notes, each being separated by a semitone. In general, the gap between whole notes (A,B etc) is a tone. The indication of whether a note goes up a semitone or down is usually shown by either a sharp (#) or a flat (♭) respectively. The exceptions (there are always exceptions) are the intervals between B & C and E & F which are semitones. The notes above show the 12 semitones No, I haven't made a mistake! Yes, more than 12 notes are shown but there are notes which are exactly the same. For example, D# and E♭ are exactly the same note.

A scale is made up of 8 notes in the following order: Tone, Tone, Tone, Semitone, Tone, Tone, Tone, Semitone. All major scales follow this series of notes. The classic scale example is the white notes on a piano, which when played in that sequence (from the letter note 'C') gives the scale of C major.

The difference between the bottom sounding 'C' and the 'C' at the top of the scale is known as an octave (8 notes). Therefore all scales start and finish on the same note value. The final note would therefore also start the next higher octave, and so on. There's no particular reasoning behind these notation conventions but it's the most generally acceptable standard which has been used in western music for hundreds of years.

Ok, enough of the music lesson. By now your fingers must be itching for some action on the keyboard so let's look at the first command in BASIC, the SOUND statement.

The Sound Command

The most terrifying part of using this command is the number of different parameters that you can use. This often deters people from using any of the sound commands because they are at first sight complicated. In fact the very opposite is true. The SOUND statement is easy to use once you clearly see the functions of each part of the statement.

Amstrad have provided facilities within the SOUND command to access the various basic elements of sound I've discussed previously, in a simple and effective manner and most importantly, directly from BASIC.

The command in all its glory looks as follows:

```
SOUND C,P,D,V,VE,TE,NP
```



As you can see the command has seven parameters but you need only use the first two. The remaining five parameters (D to NP) are optional. The first parameter, 'C' is the channel being used and its status.

This parameter has a variety of possible effects but for the moment I'm only going to look at what these sound channels are and how to play sounds through them.

If you think about music, more than one note is often being played at the same time (for example). This allows the creation of different harmonies and the same can be implemented on the Amstrad micro by allocating different channels to play different notes. The micro has 3 channels on which sound can be produced.

The second parameter, 'P' is also required and this informs the sound chip which pitch to generate the sound. Don't confuse this value with the frequency numbers given in the user guide. Always refer to the pitch number (period) for the note you wish to produce. Use the above example but change the pitch number.

One thing you may notice is that the pitch number increases as the note drops in pitch! This is correct and is a confusing feature which is easily forgotten.

The third parameter is the duration, 'D'. This is the first of the optional values which you can enter. If it is not specified any sound, as in the above example, will cause a tone to be produced for one-twentieth of a second which is the default value.

Any value between -32768 to +32767 can be entered. There are three possible ways of using this parameter. Firstly to indicate the length of time a sound is played. This must be a positive integer and each unitary value equals one-hundredth of a second. Therefore if D is equal to 100 then the sound will be played for one second.

The other two ways of using this parameter I'll leave till next time as they refer to the ENV command.

Finally for this month the fourth parameter, V, which is the volume or amplitude. In the user guide the possible values are given as between 0 to 15. However if you enter a figure between 8 to 15 you are merely repeating the effects of 1 to 7!

The amplitude of the sound increases from no volume (0) to the loudest (7).

The following program demonstrates this by stepping through each volume figure in a FOR-NEXT loop.

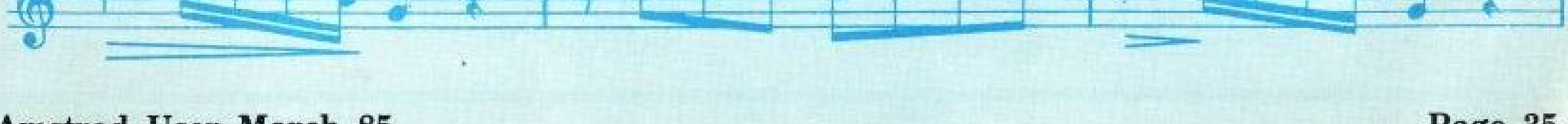
```
10 FOR volume=0 TO 7
20 SOUND 2,319,30,volume
30 NEXT
```

And that concludes this introduction to sound and music. Play around with each of the parameters discussed until you are satisfied that you understand their separate effects and next month we will start playing music on the Amstrad and look in more detail at the rest of the SOUND command and its related envelope commands, ENV and ENT.

Jeremy Vine is a freelance writer and has recently published two books on the CPC464, which includes a book on sound and music, 'Bells and Whistles on the Amstrad CPC464', published by Shiva Publishing Ltd. at £4.95p.

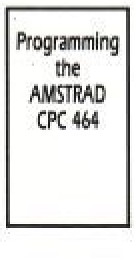
```
10 REM data format is XXX 3 bit number to say which channels
20 REM                note 2 chars to specify note
30 REM                oct  1 number to specify octave
40 REM                len  1 number to specify length
50 REM 1=sq 2=q 3=dotq 4=c 5=dotc 6=sb 8=dotsb 9=b
60 unit=4.5
70 ENV 1,unit,2,1,unit,-2,1
80 ENV 2,unit,2,1,1,0,2*unit,unit,-2,1
90 ENV 3,unit,2,1,1,0,4*unit,unit,-2,1
100 ENV 4,unit,2,1,1,0,6*unit,unit,-2,1
110 ENV 5,unit,2,1,1,0,10*unit,unit,-2,1
120 ENV 6,unit,2,1,1,0,14*unit,unit,-2,1
130 ENV 7,unit,2,1,1,0,18*unit,unit,-2,1
140 ENV 8,unit,2,1,1,0,22*unit,unit,-2,1
150 ENV 9,unit,2,1,1,0,30*unit,unit,-2,1
160 ENV 10,1,0,8*unit
170 DIM period(12)
180 period(1)=3822 : period(2)=3608 : period(3)=3405 : period(4)=3214
190 period(5)=3034 : period(6)=2863 : period(7)=2703 : period(8)=2551
200 period(9)=2408 : period(10)=2273 : period(11)=2145 : period(12)=2025
210 notes$="  c c+d d+e f f+g g+a a+b "
230 READ a$
240 WHILE a$<>"*"
250  spoint=1
260  WHILE spoint < LEN(a$)
270   chana=0: chanb=0: chanc=0
280   chan$=MID$(a$,spoint,3): spoint=spoint+3
290   IF VAL(LEFT$(chan$,1)) THEN chana=1
300   IF VAL(MID$(chan$,2,1)) THEN
      chanb=2-8*(chana<>0): chana=chana-16*(chana<>0)
310   IF VAL(RIGHT$(chan$,1)) THEN chanc=4-8*(chana<>0)-16*(chanb<>0):
      chana=chana-32*(chana<>0): chanb=chanb-32*(chanb<>0)
320   IF VAL(LEFT$(chan$,1)) THEN note$=MID$(a$,spoint,4): spoint=spoint+4:
      GOSUB 390:SOUND chana,period,0,0,envel
330   IF VAL(MID$(chan$,2,1)) THEN note$=MID$(a$,spoint,4): spoint=spoint+4:
      GOSUB 390: SOUND chanb,period,0,0,envel
340   IF VAL(RIGHT$(chan$,1)) THEN note$=MID$(a$,spoint,4): spoint=spoint+4:
      GOSUB 390: SOUND chanc,period,0,0,envel
350  WEND
360  READ a$
370 WEND
380 END
390 REM calculate note parms
400 note=INSTR(notes$,LEFT$(note$,2))
410 octave=VAL(MID$(note$,3,1))
420 envel=VAL(RIGHT$(note$,1))
430 base=period(note\2)
440 period=base/(2XXXoctave)
450 RETURN
460 DATA "101g 44g 24111d 54f 34b 24111d 54f 34b 24111e 54f 34b 24"
470 DATA "101d 54c 34111c 54g 34e 34111c 54g 34e 34111e 54g 34e 34"
480 DATA "101f 54c 34111a 54a 34f 34111c 64a 34f 34111a 54a 34f 34"
490 DATA "101a 54c 34111g 54g 34e 34111g 56g 34e 34011g 34e 34"
500 DATA "101a 54g 24111b 44f 34b 24111b 44f 34b 24111a 54f 34b 24"
510 DATA "101g 54c 34111c 54g 34e 34111c 54g 34e 34111e 54g 34e 34"
520 DATA "101e 54d 34111d 54c 44f+34111e 54c 44f+34111d 54c 44f+34"
530 DATA "101e 54g 24111d 54f+44b 24111b 44f+34b 24111g 44f+34b 24"
540 DATA "101g 44g 24111d 54f 34b 24111d 54f 34b 24111e 54f 34b 24"
550 DATA "101d 54c 34111c 54g 34e 34111c 54g 34e 34111e 54g 34e 34"
560 DATA "101f 54c 34111a 54a 34f 34111c 64a 34f 34111a 54a 34f 34"
570 DATA "101a 54c 34111g 54g 34e 34111g 56g 34e 34011g 34e 34"
580 DATA "101a 54g 24111b 44f 34b 24111b 44f 34b 24111a 54f 34b 24"
590 DATA "101g 54c 34111c 54g 34e 34111c 54g 34e 34111e 54g 34e 34"
600 DATA "101e 54g 24111d 54f 34b 24111e 54f 34b 24111d 54f 34b 24"
610 DATA "101e 54c 34111c 54g 34e 34111g 36e 36c 56"
620 DATA "101e 56c 34011a 34f 34111c 56a 34f 34011a 34f 34"
630 DATA "101a 46c 34011g 34e 34111g 46g 34e 34011g 34e 34"
640 DATA "101g 54g 24111d 54f 34b 24111e 54f 34b 24111f 54f 34b 24"
650 DATA "101e 54g 24111d 54g 34e 34111c 56g 34e 34011g 34e 34"
660 DATA "101e 56c 34011g 34e 34111c 56g 34e 34011g 34e 34"
670 DATA "101a 46c 34011g 34e 34111g 46g 34e 34011g 34e 34"
680 DATA "101f+44g 24111g 44f 34b 24111a 44f 34b 24111b 44f 34b 24"
690 DATA "101d 54c 34111c 54g 34e 34111c 56g 34e 34011g 34e 34"
700 DATA "101e 56c 34011g 34e 34111c 56g 34e 34011g 34e 34"
710 DATA "101a 46c 34011g 34e 34111g 46g 34e 34011g 34e 34"
720 DATA "101g 54g 24111d 54f 34b 24111e 54f 34b 24111f 54f 34b 24"
730 DATA "101e 54g 24111d 54g 34e 34111c 56g 34e 34011g 34e 34"
740 DATA "101e 56c 34011g 34e 34111c 56g 34e 34011g 34e 34"
750 DATA "101a 46c 34011g 34e 34111g 46g 34e 34011g 34e 34"
760 DATA "101f+44g 24111g 44f 34b 24111a 44f 34b 24111b 44f 34b 24"
761 DATA "101d 54c 34111c 54g 34e 34111c 56g 36e 36"
770 DATA "*"
```

An Example of more complex music programming.



DO MORE WITH YOUR CPC 464

Make the most of your Amstrad CPC 464 with these helpful guides — there's something here for everyone.



NO COMPUTING EXPERIENCE?

Programming the Amstrad CPC 464 is the ideal user's guide for the Amstrad owner with no experience of computing, wanting to make full use of the many facilities of this powerful machine. Author Richard Meadows introduces you to simple programming techniques, graphics, colour and sound on the Amstrad — quickly and clearly. Everything is explained simply — and there are clear program listings and exercises to help you apply and develop your skills. **£5.95** 192 pages.





THE GATEWAY TO FUN!

Gateway to Computing with the Amstrad CPC 464, Books 1 and 2

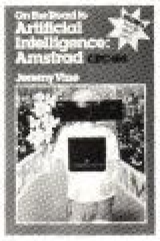
'Enter the Gateway' and discover a fun new approach to Amstrad computing when you encounter a cast of colourful characters, problems, puzzles and projects. Designed to introduce the fundamentals of computing to young people — and to adults who are still young at heart! — the books in Shiva's *Gateway to Computing* series have already proved very popular with Commodore 64, Commodore 16, BBC Micro and Spectrum owners. Now Amstrad users can enjoy the Gateway approach to computing. **Book 1** helps you to get used to your Amstrad, begin writing your own programs and learning simple instructions in BASIC. **Book 2** goes beyond elementary BASIC and enables you to develop your skills by revealing the secrets of finer programming techniques and structure. It includes plenty of sample programs. **Book 1 £4.95** 112 pages. **Book 2 £4.95** 128 pages.



MAKE MUSIC!

Bells and Whistles on the Amstrad CPC 464

Want to be a music composer, conductor, player and audience with your Amstrad? With the help of Jeremy Vine you can join in the banging, hooting and music-making and master the sound and envelope commands for this musical micro. You don't need to know anything about musical theory — and very little about BASIC either — to compose and make music, convert your Amstrad into a synthesizer or discover the inner workings of its sound chip. Learn how to make all kinds of special sound effects such as ringing telephones, explosions, alarms and sirens, musical scales and, of course, bells and whistles. Ear plugs not included! **£4.95** 150 pages.



AMSTRAD A.I.

On the Road to Artificial Intelligence: Amstrad CPC 464 is an introduction to Amstrad BASIC with a difference, because unlike other books it introduces the notion of Artificial Intelligence and shows you how to write interactive programs. Discover and develop the techniques you need to build data banks, beat the bugs, disentangle strings, establish a tamper-proof machine and allow free communication between yourself and your Amstrad. And use the two simulation programs provided to turn the CPC 464 into an intelligent companion with whom you can converse! You need very little knowledge of BASIC: and if you have *no* programming experience, don't worry — there's a crash course in BASIC at the back of the book. **£5.95** 112 pages.

Available from large branches of Boots, Menzies, W. H. Smith and bookshops, computer shops and department stores: or direct from the distributor by filling in the coupon below and sending it with your payment to HOLT SAUNDERS LTD, ACCOUNTS DEPT, 1 ST ANNE'S ROAD, EASTBOURNE BN21 3UN. (Tel. 0323 638221).

Please make cheques payable to **Holt Saunders Ltd** and allow 28 days for delivery.

*** Free postage and packing on all UK orders ***

WRITE OR PHONE TODAY FOR YOUR FREE CATALOGUE

ORDER FORM

Please send	QTY	TOTAL PRICE
Programming the Amstrad CPC 464 (ISBN 0 304 31250 9) @ £5.95 (£6.39 in Europe, £6.70 outside Europe inc. p & p)	______	______
Gateway to Computing with the Amstrad CPC 464: Book 1 (ISBN 1 85014 016 2) @ £4.95 (£5.34 in Europe, £5.50 outside Europe inc. p & p)	______	______
Gateway to Computing with the Amstrad CPC 464: Book 2 (ISBN 1 85014 023 5) @ £4.95 (£5.24 in Europe, £5.54 outside Europe inc. p & p)	______	______
Bells and Whistles on the Amstrad CPC 464 (ISBN 1 85014 063 4) @ £4.94 (£5.28 in Europe, £5.53 outside Europe inc. p & p)	______	______
On the Road to Artificial Intelligence: Amstrad CPC 464 (ISBN 1 85014 064 2) @ £5.95 (£6.34 in Europe, £6.50 outside Europe inc. p & p)	______	______

I enclose payment of £.................... by cheque ☐ or debit my credit card: Amex ☐ Diners Club ☐ Visa ☐ Access ☐ (✓)

Credit card number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Name (please print) ______________________ Address ______________________

Telephone ______________________ Signed ______________________ AU2/3

Telephone credit card sales welcome — just dial 100 (UK only) and ask the operator for FREEPHONE 2568.

SHIVA PUBLISHING LTD — Cassell COMPUTING *from* HOLT SAUNDERS

Make the most of your micro.

CP/M Assembler

The purpose of this article is to introduce you to machine code programming when running the CP/M disc operating system on your CPC464.

I shall begin by explaining the concept of a .COM file, the fundamental command file of CP/M, then show how to produce such a file using the GEN80 assembler and finally work through an extended example, an UNERAse program that recovers a file that you have accidentally deleted.

Firstly, a quick word about CP/M's memory map; whenever the renowed Control Program for Microcomputers (CP/M for short!) is running on your computer then it will have partitioned your RAM memory as follows:

`0 - #FF:` some CP/M workspace and default disc buffer

`#100 - XXXX:` the Transient Program Area (TPA for short)

XXXX upwards
: Command Control Processor (CCP)
: Basic Disc Operating System (BDOS)
: Basic Input/Output System (BIOS)

NB. XXXX is the highest byte of the TPA, #9700 on the CPC464. also the 'Basic' in BIOS and BDOS is nothing to do with the language BASIC.

Of course the casual user of CP/M sees nothing of this, the screen is displaying that ubiquitous A› prompt, waiting for you to type something. It is when you type a line (ended by ‹ENTER›) that things start moving; it goes something like this: the CCP reads in the line you have typed and looks at the first word of that line, it then searches its command table to see if it can match your word with any of its built-in commands; if it can then it obeys that command immediately. Some of the commands contained within the CCP are DIR (directory), ERA (erase a file), REN (rename a file) and TYPE (display the contents of a file).

As you can see, most of these commands expect you to specify a file (or files) on which the command will work; files are simply collections of bytes held on a disc under a particular name. CP/M stores all these names in a particular area on each disc so that it can easily see what files are on that disc, this area is called the Directory. The name of a CP/M file is split up as follows: d:nnnnnnnn.ttt

where d is the disc drive letter on which the file is to be found, nnnnnnnn is called the filename of the file (1-8 letters) and ttt is the filetype of the file (1-3 letters). The filetype is there simply for the convenience of the user, allowing you to name files of similar structure with the same, hopefully meaningful, filetype. Some examples of commonly used filetypes (or file extensions as they are sometimes known) are: TXT (text files), BAS (BASIC source files), BAK (backup files), PAS (Pascal source files) and COM (command files).

That brings us to:

What are .COM files?

Remember that CCP was scanning the line you typed in? What happens if it doesn't recognise the first word as one of its built-in commands? What it then does is to scan the directory of files on the current disc looking for a file with a filename of that first word and a filetype of COM. If it finds a file meeting these conditions then it uses the directory information for this file to load the file itself from the disc into memory. It loads the file starting at address #100 i.e. at the bottom of the Transient Program Area (TPA) and then executes address #100 i.e. it treats the file as though it were a command. This tells us what a COM file is:

a COM file is a directly executable program which is intended to run in the TPA, starting from address #100.

This command program may interface with CP/M by making calls to the BDOS to do things as write characters to the screen, read characters from the keyboard, open files, delete files, read data from files etc. The program can return to the CCP (and that A> prompt again) by RETurning, if the Stack has been preserved; or JumPing to location 0 (a RST 0 will do); which will reload CCP into high memory since it may have been corrupted by the COM program. The CCP will execute whereupon it will sit waiting for another command line.

How do we create .COM files?

The easiest way to create a command file is to use an assembler; there any many available to run under CP/M - one that will be commonly available on the CPC464 disc format is the GEN80 assembler produced by Hisoft. GEN80 accepts source from a file on disc, assembles it and produces a .COM file directly; memory is segmented into 3 buffers, one for source text, one for object and one for the symbol table and both the source and object buffers will overflow to disc thus allowing you to build up an infinitely large command file (in practice you are limited by the amount of space left on the disc).

To produce the source code (i.e. the Z80 mnemonics, assembler directives, commands etc.) you use a separate editor; you could use Wordstar, Amsword, ED (if you're desperate) or Hisoft's own screen editor called ED80. Then submit the source text to the assembler by typing something like:

```
A>GEN80 TEST;4;20;15
```

which means: load the GEN80 assembler (itself a COM file) which then takes the filename TEST from the command line, assumes a filetype of .GEN and assembles the source text. The numbers after the filename separated by semi-colons are:

```
;4   option meaning don't produce an assembly listing
;20  allocate a source buffer of 20K
;15  allocate a symbol table buffer of 15K
```

The assembler then reads some or all of the text into its buffer, assembles, reading another input buffer full if necessary, and produces the object code (i.e.Z80 machine language) in another buffer which it writes to disc whenever the output buffer is full or the assembly finishes. The default filename for the object code will be TEST.COM.

Linkage

This direct production of a COMmand file from the source text file in one step is not common to the working of all CP/M assemblers, many produce an intermediate file which must then be submitted to another program, called a 'linker', possibly together with other intermediate files to produce the final COM file. This process was developed to enable the production of large COM files at a time when editing large source files was both slow and unwieldly (because of small memories. slow discs and inadequate editors). Nowadays it is possible to edit large source files easily and quickly and thus the idea of object code linkage is not as advantageous as it used to be. However, there is the advantage that sub-programs can be developed and tested separately and then linked together in one final stage. Obviously, references to variables in other modules must be handled somehow and this is done through the concept of GLOBALs and EXTERNALs: globals are labels which are declared within the module to be used by another module whereas externals are labels used within the current module but declared in another module. The linker satisfies all these cross references to produce the executable command file.

GEN80 does not support object code linkage but rather allows source code linkage i.e. the concept of assembling several different source files in one assembly session to produce a COM file. This is simpler and quicker to use than object code linkage but requires a different discipline in the production of the source text files.

Another commonly supported feature of CP/M assemblers is macros.

Macros

The principal reason for using macros is to make your program smaller, easier to write and easier to understand. A macro is a sequence of assembly language instructions that may be invoked through the simple use of the macro name. An example should clarify this:

```
; Macro definition
MOVE    MAC
        LD      BC,&1-&0+1
        LD      DE,&2
        LD      HL,&0
        LDIR
        ENDM
; Macro invocation
        MOVE    128,255,4096
```

The macro definition defines a sequence of instructions that may refer to parameters (in this case using an '&' symbol) and, in some assemblers, may even refer to labels defined within the macro. The macro invocation MOVE 128,255,4096 causes the assembler to generate the following code:

```
LD    BC,128         ;255-128+1
LD    DE,4096
LD    HL,128
LDIR
```

which moves the bytes at 128 to 255 inclusive to location 4096 onwards.

Thus you can continually use the MOVE statement as if it were a new and powerful mnemonic within the Z80 set conveying more meaning to your program while making it easier to write and document. You must be careful, though, not to use macros when a subroutine would do; a macro invocation always generates the code within the macro definition and thus could result in duplication of code that should be in a subroutine.

The above is of necessity a short discussion of the use of macros, many other features (such as nested macro calls and definitions) may be supported by the assembler allowing almost high-level language programming techniques to be employed.

GEN80 supports simple, easy-to-use macros passing parameters by value as in the above example.

I hope that the above article has given you some inkling of how assemblers are used with the CP/M operating system and has removed some of the mystery surrounding the production and use of COM files.

An Example Program

To complete the article, I now present the source listing of a useful CP/M utility program. The code, when assembled, is an 'unerase' program allowing you to recover a CP/M file that you have just accidentally erased (perhaps using the CCP's ERA command). The command should be used immediately after the file has been deleted, otherwise some of the space on the disc used by the deleted file may have been re-used.

To assemble the command with GEN80 you could use:

```
A>GEN80 UNERA;5
```

to produce the command file UNERA.COM. Subsequently to unerase a file called OOPS.XXX you could type:

```
A>UNERA OOPS.XXX
```

A few notes about the program:

Subroutine Dos at line 150 is a general interface to the various standard CP/M function calls that perform tasks such as writing characters to the screen, reading records from a file etc. The function number is passed in register C e.g. LD C,1 :CALL Dos reads a character (into register A) from the keyboard.

The CP/M directory consists of a 32 byte entry for each 16K block (an extent) of each file. The first byte of each entry is the user number, in the range 0-15 (#0-#F), which was active when the file was saved (this is usually 0 unless changed by the user) if the file is present on the disc or #E5 if the file is erased. The filename and filetype of the file then follow in the next 11 bytes of the directory entry, followed by information relating to the size and whereabouts of the file.

software that's hard to beat...

AMSTRAD CPC 464

FROM THE COMPANY THAT PRODUCED AMSWORD/TASWORD 464 ("THERE IS NO BETTER JUSTIFICATION FOR BUYING A 464 THAN THIS PROGRAM" – POPULAR COMPUTING WEEKLY, 1ST NOVEMBER 1984) TWO ADDITIONAL PROGRAMS FOR YOUR CPC 464 AND DOT-MATRIX PRINTER.

TASPRINT 464 The Style Writer

A must for dot-matrix printer owners! Print your program output and listings in a choice of five impressive print styles. TASPRINT 464 utilises the graphics capabilities of dot-matrix printers to form, with a double pass of the printhead, output in a range of five fonts varying from the futuristic DATA-RUN to the hand-writing style of PALACE SCRIPT. TASPRINT 464 drives the dot-matrix printers listed below and can be used to print AMSWORD/TASWORD 464 text files. TASPRINT 464 gives your output originality and style.

FIVE IMPRESSIVE PRINT STYLES FOR YOUR USE

COMPACTA – bold and heavy, good for emphasis
DATA-RUN → A FUTURISTIC SCRIPT
LECTURA – clean and pleasing to read
MEDIAN – a serious business-like script
Palace Script – a distinctive flowing font

PRICE £9.90

TASCOPY 464 The Screen Copier

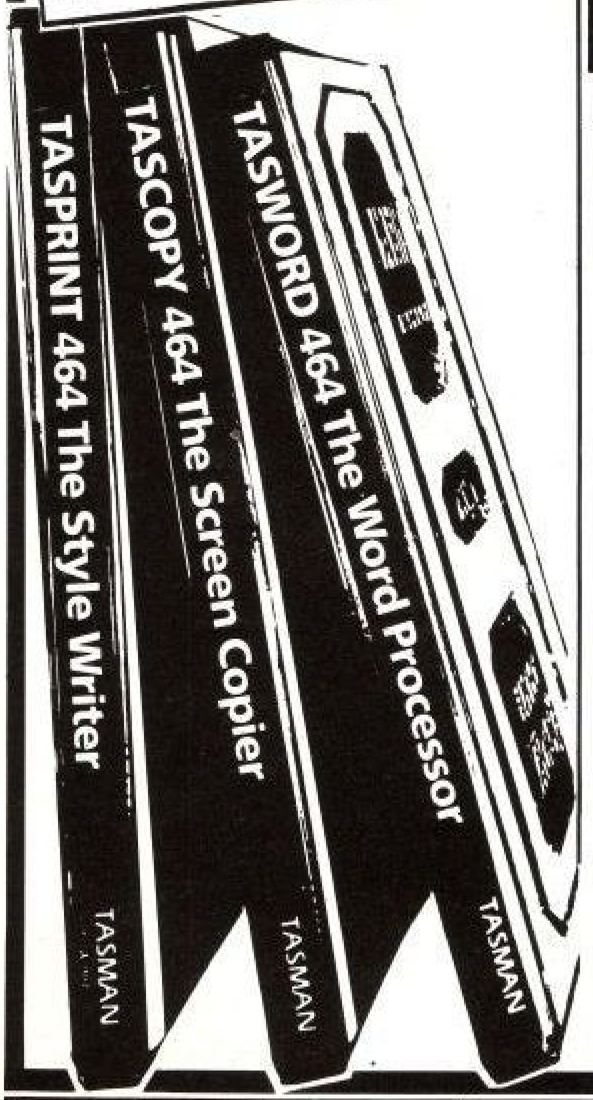
A suite of fast machine code screen copy software for the CPC 464. Print high-resolution screen copies in black and white and also large 'shaded' copies with different dot densities for the various screen colours. TASCOPY 464 drives the dot-matrix printers in the list below.

TASCOPY 464 The Screen Copier

only **£9.90**

TASPRINT 464 and TASCOPY 464 drive the following dot-matrix printers:

Amstrad DMP 1
Epson FX-80
Epson RX-80
Epson MX-80 Type III
Mannesmann Tally MT-80
Star DMP 501/515
Brother HR5



Available from good software stockists and direct from:

Tasman SOFTWARE

Springfield House, Hyde Terrace, Leeds LS2 9LN. Tel: (0532) 438301

If you do not want to cut this magazine just write your order and post to:
TASMAN SOFTWARE, dept. CPC., Springfield House, Hyde Terrace, Leeds LS2 9LN.
I enclose a cheque/P.O. made payable to Tasman Software Ltd. OR charge my

ACCESS/VISA number ________
NAME ________
ADDRESS ________

ITEM	PRICE
________	£ ____
________	£ ____
________	£ ____
TOTAL	£ ____

Send me the free Tasman Brochure describing your products for the Amstrad CPC 464 tick here: ☐

VISA

```
              *H Unerase a file   4/11/84 ·

              ;Copyright David Link 1984

              ;A program to unerase a file that has been accidentally
              ;erased. Should be used immediately after erasing the
              ;file since if used later, some blocks may have been re-used

              ;Format is: UNERA filename


005C          DEFCB       EQU   #5C

0008          fnamelen    EQU   8
0003          extlen      EQU   3
000C          extent      EQU   12
0020          dirlen      EQU   32


              ;CP/M BDOS call numbers

000F          OPEN        EQU   15
0010          CLOSE       EQU   16
0011          SEARCH      EQU   17
0012          SRCH_AGAIN  EQU   18
0016          MAKE        EQU   22
001A          SETDMA      EQU   26

              ;Default workspace for file reads

0080          tbuff       EQU   128


              ;Macro to call CP/M setting DE and C.

0100          DOS         MAC
0100                      LD      DE,=0
0100                      LD      C,=1
0100                      CALL  Dos
0100                      ENDM

              ;Macro to call CP/M setting C.

0100          SDOS        MAC
0100                      LD      C,=0
0100                      CALL  Dos
0100                      ENDM


              ;COM files begin at #100

0100                      ORG     #100

0100 ED7B0600             LD      SP,(6)              ;set stack to top of TPA
0104                      DOS     DEFCB,SEARCH        ;Does file exist?
010C 3C                   INC     A
010D C20000               JP      NZ,0                ;File exists, return to CCP
0110 21C301               LD      HL,FCBSPACE         ;initialise pointer to current FCB
0113 229D01               LD      (FCBPTR),HL
0116                      DOS     tbuff,SETDMA        ;set disc I/O to tbuff
011E                      DOS     DUMFCB,SEARCH       ;and search for first entry in directory
0126 3C       More_Searc  INC     A
0127 2840                 JR      Z,End_of_Directory ;no more entries

              ;Compare found filename with the required filename

0129 3D       Continue    DEC     A                   ;adjust because we INCed it
012A 87                   ADD     A,A                 ;multiply by dirlen to get position
012B 87                   ADD     A,A                 ;of entry in cataloque
012C 87                   ADD     A,A
012D 87                   ADD     A,A
012E 87                   ADD     A,A
012F 1600                 LD      D,0
0131 5F                   LD      E,A
0132 218000               LD      HL,tbuff            ;point to found file
```

```
0135 19                         ADD     HL,DE
0136 E5           Again         PUSH    HL                  ;and compare it with required filename
0137 23                         INC     HL
0138 060B                       LD      B,fnamelen+extlen   ;both name and type (8,name : 3,type)
013A 115D00                     LD      DE,DEFCB+1          ;filename starts at FCB + 1
013D 1A           Match         LD      A,(DE)
013E 4E                         LD      C,(HL)
013F CBB9                       RES     7,C                 ;some CP/M's set bits on filename
0141 B9                         CP      C                   ;so make sure top bit is reset for comparison
0142 23                         INC     HL
0143 13                         INC     DE
0144 2018                       JR      NZ,NoMatch          ;no this one
0146 10F5                       DJNZ    Match               ;good so far....keep going

                                                            ;Match found

0148 E1                         POP     HL                  ;filename match but.....
0149 E5                         PUSH    HL
014A 7E                         LD      A,(HL)              ;...is it erased?
014B FEE5                       CP      #E5                 ;#E5 in first byte of directory = erased
014D 200F                       JR      NZ,NoMatch          ;not erased...so search some more
014F 3600                       LD      (HL),0              ;it was erased, so unerase it
0151 ED5B9D01                   LD      DE,(FCBPTR)         ;and store the FCB information in
0155 012000                     LD      BC,dirlen           ;our temporary table
0158 EDB0                       LDIR
015A ED539D01                   LD      (FCBPTR),DE         ;updating table pointer afterwards

                  ;Search for another entry

015E E1           NoMatch       POP     HL
015F                            DOS     DUMFCB,SRCH_AGAIN   ;search for next entry in directory
0167 18BD                       JR      More_Search


                  ;We have now exhausted the directory search and built
                  ;up our table of directory entries for the required file

                  End_of_Dir

0169 11C301                     LD      DE,FCBSPACE         ;start processing table of matched
                                                            ;directory entries
016C 2A9D01       MAIN_LOOP     LD      HL,(FCBPTR)         ;have we reached end of table?
016F B7                         OR      A
0170 ED52                       SBC     HL,DE
0172 CA0000                     JP      Z,0                 ;finished so return
0175 D5                         PUSH    DE                  ;save table pointer
0176 210C00                     LD      HL,extent           ;address extent byte
0179 19                         ADD     HL,DE
017A 7E                         LD      A,(HL)              ;make an extent of new,
017B 326800                     LD      (DEFCB+extent),A    ;unerased file....
017E                            DOS     DEFCB,MAKE
0186 D1                         POP     DE                  ;recover pointer to FCB....
0187                            SDOS    CLOSE               ;....and close it
018C 212000                     LD      HL,dirlen
018F 19                         ADD     HL,DE               ;have we reached end of table?
0190 EB                         EX      DE,HL
0191 18D9                       JR      MAIN_LOOP
0193 E5           Dos           PUSH    HL
0194 D5                         PUSH    DE
0195 C5                         PUSH    BC
0196 CD0500                     CALL    5
0199 C1                         POP     BC
019A D1                         POP     DE
019B E1                         POP     HL
019C C9                         RET
019D              FCBPTR        DEFS    2
019F 3F3F3F3F     DUMFCB        DEFM    "???????????"
01AB 00000000                   DEFW    0,0
01AF                            DEFS    16
01BF 00000000                   DEFW    0,0
01C3              FCBSPACE      EQU     $
```

Have you got the drive to succeed in business?



£199

DD1-1 disc drive (with CP/M* and DR. LOGO*) and interface.

£159

FD-1 additional disc drive.

Running any business successfully means first running it efficiently.

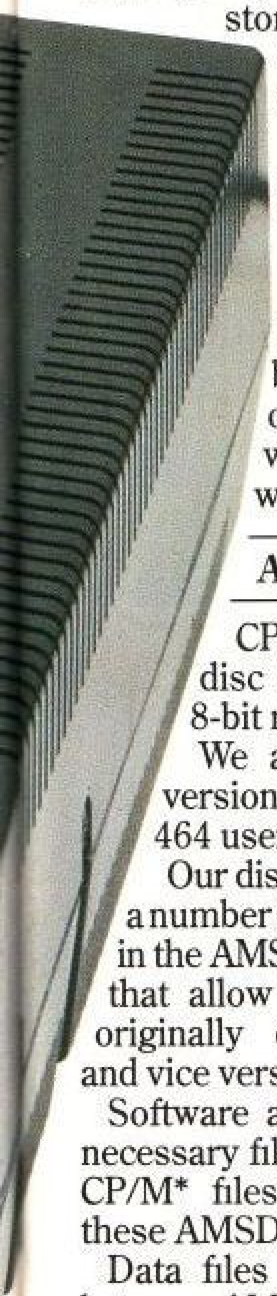
And no matter how business-like you think you are, you can't beat the power and convenience of a random access CP/M* disc system.

Now, with Amstrad's CPC 464 computer, you can enjoy all the advantages of a 3" disc drive complete with an integral power supply and plug-in interface controller.

Naturally, you can add a second drive (FD-1) to double the on-line storage capacity, speed up copying files and producing back-up discs.

But of course, simply plugging a disc drive into a computer won't get you very far.

That's why Amsoft have produced a disc based software range of over 30 programs with many more on the way.



£239

complete with green screen VDU (GT64) and FREE software pack worth over £100.

£349

complete with colour monitor (CTM640) and FREE software pack worth over £100.



CPC 464 complete with monitor and datacorder

AMSDOS and CP/M*

CP/M* is *the* standard disc operating system for 8-bit microcomputers.

We also supply a special version of DR. LOGO* for CPC 464 users.

Our disc drive system includes a number of extensions to BASIC in the AMSDOS operating system that allow the disc to use files originally created for cassette, and vice versa.

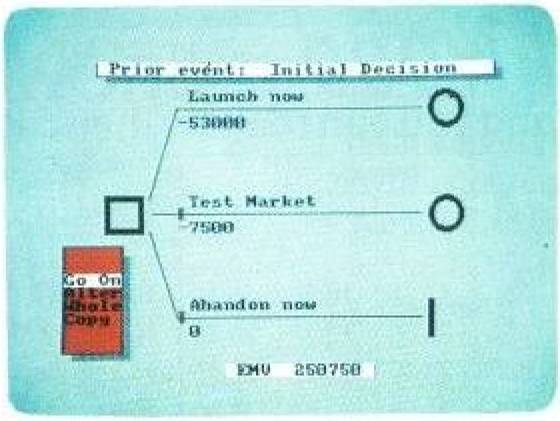
Software also takes care of the necessary file management so that CP/M* files can exist alongside these AMSDOS files.

Data files may even be shared between AMSDOS and CP/M*

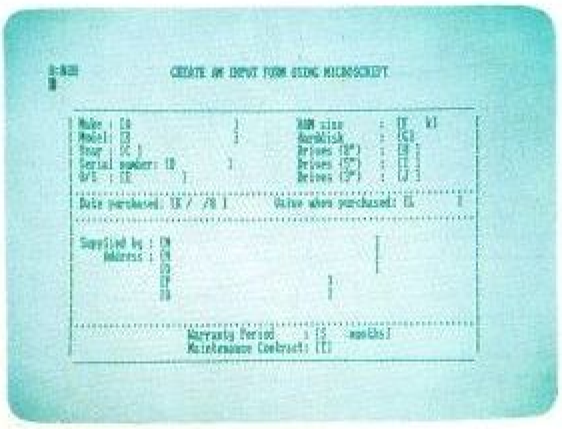
Word processing, data management, accounting, ledgers – you'll find an AMSOFT program to suit whatever kind of business you want to do on your CPC 464.

There's even a range of programs that teach you professional business practice.

Incidentally, you can exchange your Amsoft cassettes for identical Amsoft disc software for a mere £4.95 per cassette (the cost of a blank disc).



Decision Maker



Microscript

CPC 464.
Really gets down to business.

At the heart of it all, of course, is the incredible Amstrad CPC 464 computer.

The CPC 464 has a typewriter-style keyboard, large ENTRY key, sensibly positioned cursor keys, numeric keypad for fast data entry and a full 8-bit character set.

It provides high resolution graphics, 80 column text display, up to 8 text windows plus a graphics window and a palette of 27 colours.

There's also a built-in Centronics standard 7-bit parallel printer interface. So you can enjoy high performance word processing with the printer of your choice.

The CPC 464.

It does whatever you want it to do. But that, of course, is your business.



Optional 80 column dot matrix printer DMP-1. Operates at up to 50 characters per second.

AMSTRAD

ONE GREAT IDEA AFTER ANOTHER

BOOTS COMET Currys Dixons Greens John Menzies RUMBELOWS spectrum WHSMITH WIGFALLS AND OTHER COMPUTER STORES

*CP/M and DR. LOGO are Trade Marks of Digital Research

I'd like to know more about the businesslike CPC 464 complete computer system. Please send me literature right away.

NAME ______________________

ADDRESS ______________________

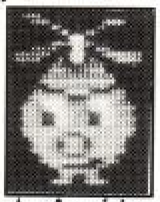
POSTCODE ______________________

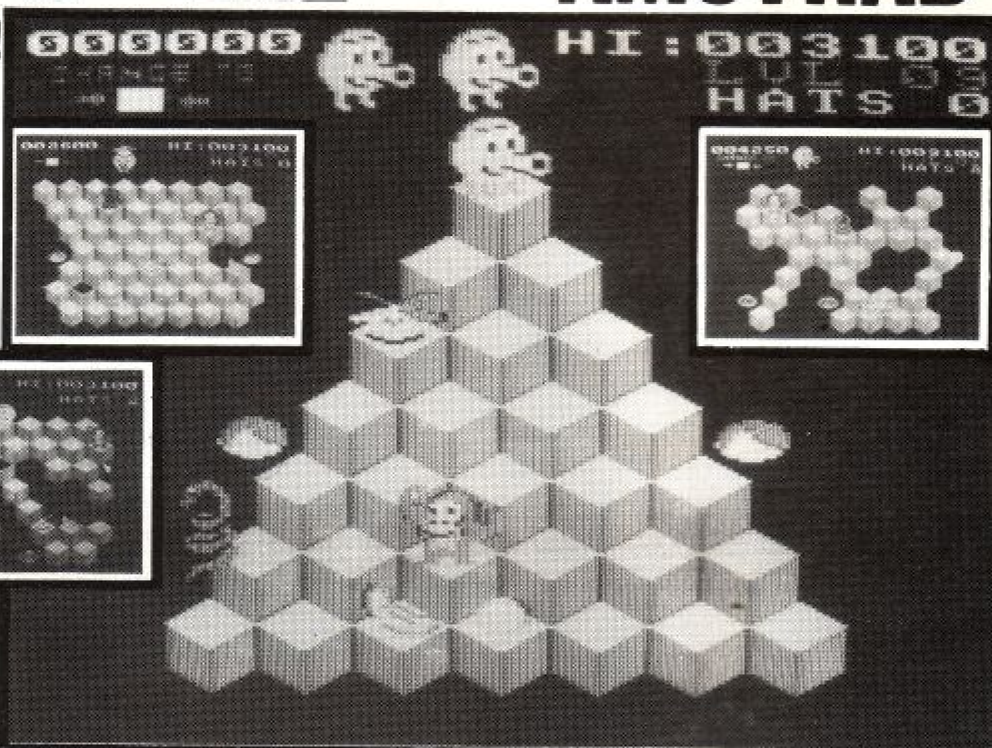
To: Amstrad Consumer Electronics PLC, Brentwood House, 169 King's Road, Brentwood, Essex CM14 4EF.

PRICES CORRECT AT TIME OF GOING TO PRESS. NOT ALL STORES STOCK ALL PRODUCTS.

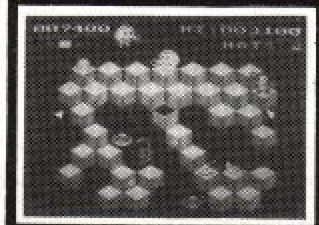
NOW! ER*BERT'S 'ERE for the AMSTRAD

IT'S ER*BERTS CUBIC DOMAIN! FAST - FUNNY - ADDICTIVE.





★ Avoid his unwelcome guests
★ Grab the banana for double scores -but watch out for Boris -he'll soon want it back! Drop it and run -unless you are very brave.
★ Avoid cascading balls - moving hole -and coily the snake - he's got a cuddle you'll never forget.
★ Escape when it gets tough - by transporter disc or rota hat, but only if you've earned one.
★ Multiple screens - additional change cube colour tasks.
★ Its fun at level one - but watch out at level ten! Packed with excitement.



ER*BERT Machine Code Game **£5.95** Inc VAT & Postage

Microbyte Software (M1),
18 Hilgrove Road, Newquay, Cornwall, TR7 2QZ

MICROBYTE SOFTWARE

Available now by fast mail order, direct from Microbyte Software.
or
Ask your dealer to stock ER*BERT today!

ACCESS CARD WELCOME
24 HR HOT LINE:
06373 6886

Dealers call 06373 6886 for Trade Enquiries

Coming Soon.....
3D SPACE RANGER

THE Space game to challenge your SKILL!

Available for the Amstrad January 1985



BIGGER STOCKS

GARWOOD IS GROWING!
(and you can reap the benefit)

CHEAPER PRICES

DISKS				DYSAN		3M		MEMOREX		BASF		HITACHI	
Price per box of 10			TPI	1-4	5-9	1-4	5-9	1-4	5-9	1-4	5-9	1-4	5-9
5¼"	S/S	S/D	48					15.60	14.70	14.60	13.70		
	S/S	D/D	48	17.25	16.20	15.10	14.20	17.10	16.00	17.60	16.50		
	D/S	D/D	48	25.30	23.70	20.70	19.40	20.60	19.30	21.50	20.20		
	S/S	Q/D	96	25.30	23.70	23.10	21.80	24.70	23.20	23.10	21.70		
	D/S	Q/D	96	31.80	29.80	27.00	25.30	27.40	25.70	26.80	25.20		
8"	S/S	S/D	48			16.80	15.75	19.80	18.50	15.20	14.30		
	S/S	D/D	48	24.70	23.15	21.75	20.40	20.80	19.50	20.00	18.80		
	D/S	D/D	48	29.30	27.45	25.05	23.50	27.10	25.40	23.80	22.30		
3" Compact (for the Amstrad Disc Drive)												36.00	33.80
3½" Micro	S/S	D/D	135					41.80	39.20				

Disc prices are exc. VAT but inc. carriage. S/S = Single Sided D/S = Double Sided S/D = Single Density D/D = Double Density Q/D = Quad Density

RIBBONS supplied for most printers **(including the Amstrad DMP1)** Prices on Application

LISTING PAPER (plain or ruled) Priced per 1000 forms			Boxed	1-4 boxes	5-9 boxes	10+boxes
11" × 8½"	1PT	60 gsm	2000	4.95	4.80	4.65
11" × 8½"	2PT	NCR	1000	14.45	13.95	13.45
11"×9½"	1PT	60 gsm	2000	5.25	5.10	4.90
11"×9½"	1PT	70 gsm	1750	6.00	5.80	5.60
11"×9½"	2PT	NCR	1000	15.80	15.25	14.75
11"×14½"	1PT	60 gsm	2000	7.30	7.05	6.80

Listing Paper prices are exc. VAT. Delivery free within 20 mile radius of Brentwood. Carriage at cost outside this area.

PRINTER OFFER - MT80 Matrix Printer 80 c.p.s. RRP £217 + VAT OUR PRICE **£199** + VAT

COMPUTER COMPANIES formed with your choice of name - **£85** + VAT (NO EXTRAS)

GARWOOD's STATIONERY CATALOGUE - Full colour 224 pages - **£1.75** inc postage

Garwood (Wholesale) Ltd
45 Plovers Mead, Wyatts Green, Essex CM15 0PS ☎ Blackmore (0277) 823747

Win!

Six Speech Synth's

DK'Tronics have won the race to be the first company to bring out a speech synthesiser for Arnold. It costs £40 but you could have one for nothing. Amstrad User have six of the devices to give away. All you have to do is look at the two pictures on this page and spot as many differences as you can. Just to make it a little bit harder we won't tell you how many there are. The competition is open to everyone, not just members of the user club. With the speech synth' plugged into the back of your computer you can make it talk with the minimum of programming. The software which is included provides a set of RSX's to control the speech. There are two speakers with the unit to provide stereo output for the music from your computer. The voice is still mono but it is clear and easily understandable. A full review will appear in the next issue.



RULES:

1 The winners will be the first six correct entries drawn on March 29th 1985.
2 There is no limit to the number of entries any individual can make, however all entries must be on an official order form, photocopies are not acceptable.
3 Entries should be sent to:
Amstrad User Magazine
Speech Synth. Comp.
169 Kings Road
Brentwood
CM14 4EF
4 No correspondence can be entered into regarding the competition. Entries cannot be returned.
5 The judges' decision is final, no employees of Amsoft, Amstrad, Amstrad User or their relatives may enter.
6 Every entry must have the number of differences written on the back of the enevelope.

How to Win.

At the bottom of this page you will see two simillar pictures. All you have to do is identify the differences in the bottom picture and ring round them in pen. When you think you have found all the differences, fill in the entry form and send it to us. Don't forget to write the number of differences on the back of the envelope.. Happy counting.



Official Competition Entry Form

AMSTRAD CPC 464 USER

Name ....

Address ....

....

....

Number of differences found .... [ ]

ZEN and the Art of Machine code Programming

The Amstrad user who wants to progress beyond BASIC into assembly language is really spoiled for choice, because in addition to the excellent HiSoft "Devpac" marketed by Amsoft (and already reviewed), Kuma Computers have now made available their equally good ZEN assembler for the Amstrad CPC464.

As you probably know, the only language your computer really understands is machine language (ML), a collection of figures in hexadecimal notation that can be very daunting for the beginner. Inside the computer is an interpreter which translates your BASIC into machine language. Every BASIC instruction has to be interpreted into ML before it can be executed, which is why Basic programs run slower than those written in ML.

To enter ML, you need a monitor, since the Amstrad CPC464 doesn't have one resident. ML programs are often published in computer magazines, but as anyone who's tried typing in magazine listings will tell you, typing errors are inevitable, and unless you really understand what you're doing, finding the mistyping is almost impossible. Chances are you'll also "hang" to machine. (Don't worry: though it's alarming, a "hang" won't damage the machine, but you'll have to reset it to start again.)

But anyway, typing in all those numbers and letters without the faintest idea what you're doing can be deadly boring.

The solution is an "assembler", which is a sort of in-between programming language, not so easy on the eye as BASIC, but not so dense as ML. It uses fairly easy-to-understand mnemonics, like LD for "load", to create a text file of what's known as source code. Then it translates this into what's known as object code, which is actually our new friend ML. The ML can be SAVEd, LOADed, RUN like any other program (though it doesn't use those actual commands).

Kuma's ZEN is a fine example of what can be done with an assembler. It's well documented, as far as it goes, and works well. It's not very much help to the absolute beginner, however, because there are no examples or exercises to work through. This is because it's really aimed at someone with a bit of experience of Z80 assembler.

ZEN comes on as a binary file running at address 4000H (16384 decimal) and the memory must be lowered to protect it with the command.

`MEMORY 16383`

followed by

`LOAD 'ZEN'` **[ENTER]**

After LOADing, you are still in BASIC, and control must be transferred to ZEN with:

`CALL 16384` **[ENTER]**

The opening menu tells you one of the nicest things about ZEN: that it uses easily memorable single-character commands, which barely need any explanation:

A - Assemble	O . Out
B - Bye	P - Print
C - Copy	Q - Query
D - Down	R - Read
E - Enter	S - Sort
F - Fill	T - Target
G - Goto	U - Up
H - Howbig	W - Write
I - In	X - eXamine
K - Kill	Z - Zap
L - Locate	c - catalog
M - Modify	d - disassemble
N - New	u - unscramble

Mistypings can be corrected with the **[DEL]** key. If ZEN doesn't understand a command, it gives you the friendly error message:

`HUH?`

Pressing **[ENTER]** on its own will clear the screen.

Catalog is the same as the BASIC `CAT` command, verifying and displaying all the files found on the cassette, indicating whether they are BASIC ($), protected (%), Binary (&), or text (*).

The "disassemble" instruction is interesting, because it allows you to examine the way programs work -- yes, including ZEN itself, and the Amstrad's Basic commands in ROM. They can be LISTed to the screen or to other external device like a printer.

The "unscramble" is similar, but only looks at eight Z80 instructions. For instance, if you want to look at the start of ZEN's main loop type:

`u41E0H` **[ENTER]**

(the lower case "u" is essential, to avoid confusion with Up).

Though it comes first alphatically, "Assemble" is actually the final command you'll use, after you've finished putting together your "source code", a text file consisting of the Z80 mnemonics, and want to produce the "object code" in machine language, which is what you actually `SAVE` (though in this assembler you need to use the command Write). As in "disassemble", you can have it display what it's doing either on the screen, printer or other device. As you might expect, this generates reams of stuff, which you don't have to see if you don't want, because there's a "null" output option. This speeds up the operation considerably.

"Bye" is the next command you'll need, returning you to Basic when everything's done. If you do it in error, don't worry, because you can get back to where you were with the `CALL 16384` command as before, and any files or data will still be in memory.

"Copy" moves a block of memory. You have to define the START and STOP parameters and the DESTINATION you want it moved to; the value may be decimal, hexadecimal (with an "H" at the end) or octal ("O" at the end). No suffix assumes a decimal figure.

Some assemblers use line numbers for help in finding your way around (though line numbers really have no meaning away from BASIC) but in ZEN on the Amstrad, line numbers are not displayed because of the 40-characters screen width. Commands like "Up" and "Down" move the appropriate number of lines in the appropriate direction, so U30 will display the line 30 lines above the previous position.

You start typing in your instructions with the command "Enter", which allows you to type in lines of text, each terminated with **[ENTER]**, until a full stop is typed in as first character of line. "Enter" can be used at any time, and will move the previous lines down to make room for the new lines.

"Howbig" lets you know where your ZEN text file starts and ends, and where is the top of memory. "Kill" erases the text file and is the equivalent of NEW in Basic, but "New" in assembler doesn't mean the same thing: it lets you edit the current line.

Likewise, "Locate" has nothing to do with the cursor-positioning command used in Locomotive BASIC: it is really a "Find" to look for the first ocurrence of a string. (Presumabley, they'd have used "Find" if they hadn't needed "Fill" to fill a block of data with a given DATA value.)

"Modify" is like "New", but allows you to define the start address in memory that you want to examine and edit. "Print" displays the next line(s) in memory of the screen, depending how many lines are specified. One line is the default. "Query" is similar, but it displays 64 bytes of memory in HEX and ASCII, from the current line if no parameter is specified.

"Read" is the assembler equivalent of LOAD and the user will be prompted for a file name. A text (Basic) file will be appended to the end of any text in memory. If it is too large an error message is displayed and Reading is terminated.

Binary files are normally Read into their load address, unless a specifically different load address has been defined.

Similarly, "Write" is the equivalent of SAVE, with W on its own writing a Basic text file, and WB will write a binary file. Start, stop and exec addresses will be requested in the latter case.

ZEN is well documented, containing a full Z80 instruction set, an assembly-language listing of ZEN itself, plus an alphabetical index of where all its instructions can be found, and a clear, concise guide to each of its special instructions.

It is a tool, and a good tool, but just as you wouldn't let a toddler loose with a wood chisel, it needs care and understanding if you're to get the full use out of it. The lack of any worked examples may hamper the absolute beginner who wants fast access to some of the more sophisticated facilities of the Amstrad, but then that never was its intention.

Some indication of how to drive the sound and graphics firmware of the Amstrad via the Z80 might have been useful, however, since the increased speed of ML games is one of the best reasons many users will have for getting into assembler.

Some of this info is in Kuma's "Amstrad Explored" book, but a specific book on this subject is said to be on the way.

One good thing is that you can experiment as much as you like, but as long as you don't Write over the program cassette any modifications you might try to ZEN, you can't do any damage either to the program or your computer, though you might hang it up, and would have to turn everything off and on to get back to square one.

If you are serious about wanting to get into assembly language, as well as this program you'll need a good guide to the Z80 chip and its instruction set, either Rodney Zaks' 624-page "Programming the Z80" (Sybex, £13.95) or the almost equally-massive, but less forbidding "Z80 Assembly Language Programming", Lance A Leventhal (Osborne/McGraw Hill, £13.95). I'd also advise Amsoft's own "Complete CPC464 Operating System Firmware Specification" (Soft 158).

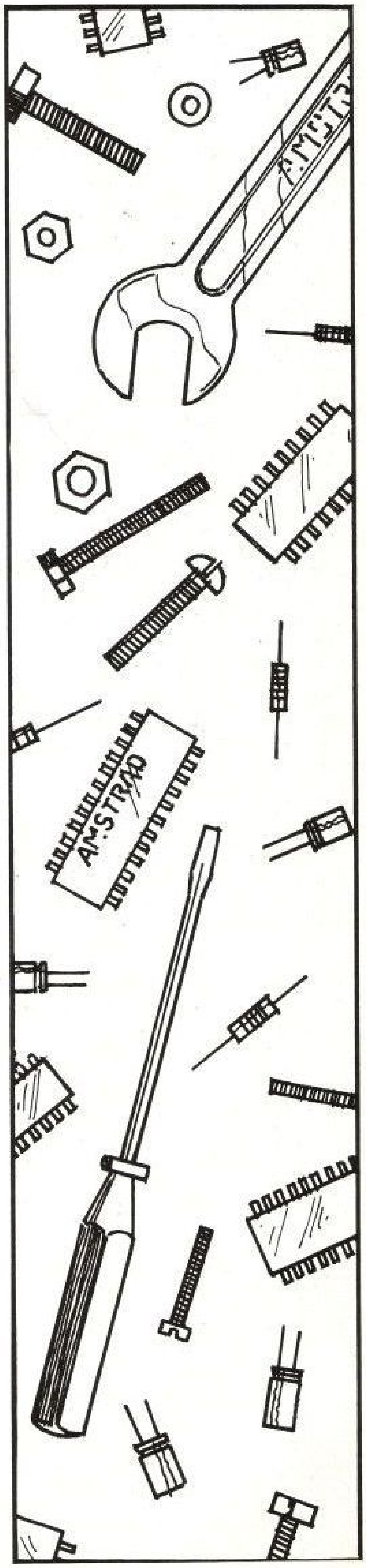
CONCLUSIONS

ZEN Z80 Assembly Language Programming System (Kuma, £19.95 inc VAT)

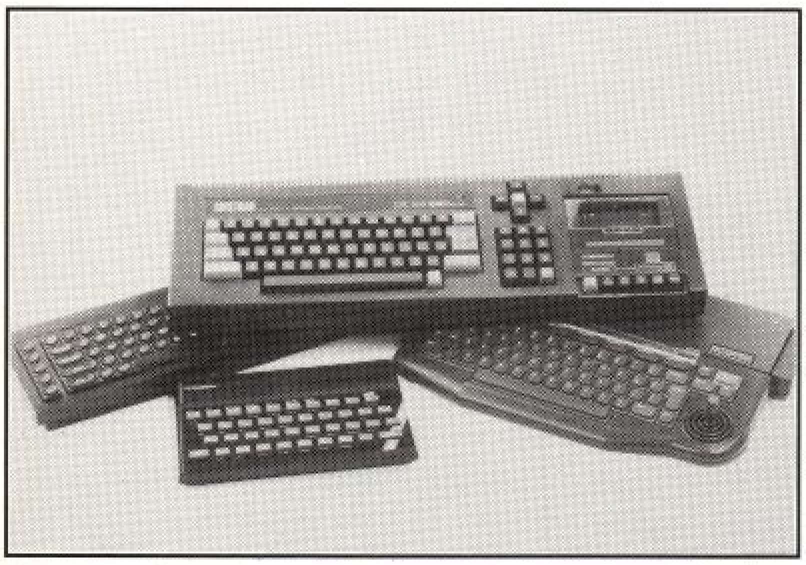
Supplied on cassette

Manual: A5, 10pp introduction plus Z80 instruction set, ZEN assembler listing, and alphabetical list of opcodes. If you're a serious programmer, then ZEN will be an essential utility for you. If you're a beginner, it may be a bit off-putting, but worth the trouble.

Karl Dallas



High-Speed Locomotive



The CPC 464 shows a clear pair of heels to it's rivals

I was reading a book on the language 'C' the other day, and came upon a program that the author was using as to show the speed of 'C' compared to Basic. In case you haven't heard of it, C is one of the newer compiled languages. Because it is compiled, that is turned into a block of machine code, before it is run, a C program will run faster than a BASIC program, where the computer has to re-interpret every line each time it meets it.

The example given was the Sieve of Erasosthenes - a means of filtering out prime numbers. You start with a block of numbers, and mark off all the mulitples of 2, from 4 onwards, then all the multiples of 3 (from 6), 5, 7,11, etc. As you work through the block, each time you meet a number which hasn't been marked off, you know this must be the next prime number, and then go off and cast out all of its multiples.

According to Bruce Hunter, author of 'Understanding C', a program to find the primes on 8K's worth of numbers, (0 to 8191) would take about 15 seconds in C. He goes on to say, "Try the same thing on any BASIC interpreter - when you have all day to watch it run. It should be about as exciting as watching paint dry. " Well, I did have some spare time the other day, so I tried it on the Amstrad CPC464. The program I used is shown here. It is the equivalent of the C program from the book. The block of numbers is an array - `a%(8191)`. The outer loop takes the program through the array, selecting those numbers that have not been marked. The start-finish times are included for the benefit of those of us that don't own stopwatches.

Run the program, and it does seem rather slow at first. It takes a few seconds to cast out the multiples of 2 and 3, but it soon speeds up. In all, the program takes a little over 100 seconds to run - about 7 times as long as the C program. However, the comparison is not quite fair, as the C original did not print out primes as it ran, so delete the second part of line 70 and try again. Definitely not as interesting to watch, but if you listen carefully you can hear the machine thinking! It doesn't take so long either, about 67 seconds, or about 4 times the speed of C. Not bad for a supposed slow-coach!

Just for interest, I tried exactly the same program on a Spectrum and on a Commodore 64. The Spectrum managed the printing version in 332 seconds, and the same on the Commodore took 352 seconds. So, perhaps Mr. Hunter wasn't so wrong after all, it was just that when he wrote his book he had never heard of Locomotive Basic.

```
10 start=TIME
20 DIM a%(8191)
30 FOR n%=2 TO 8191
40 IF a%(n%) THEN 80
50 FOR t%=n%+n% TO 8191 STEP n%
60 a%(t%)=1
70 NEXT:PRINTn%
80 NEXT
90 finish=TIME-start
100 PRINT finish/300
```

Run time = 102.353 seconds

PERSONALISED HANDMADE JUMPERS TO ORDER

Select your favourite character from **ANY** Amsoft game to feature on a truly unique jumper.



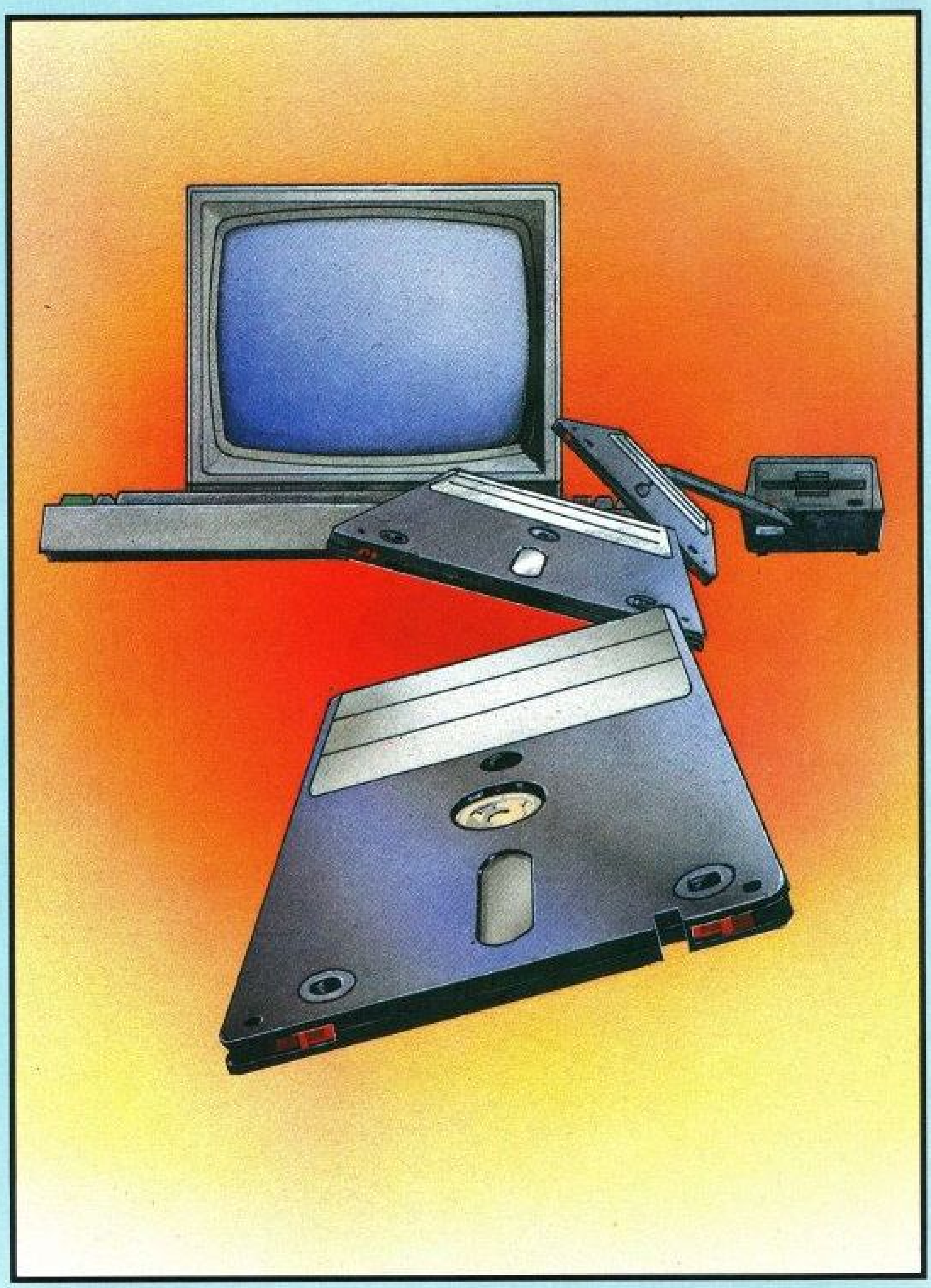
Any orders considered

For further details and prices write to

Common Denominators Ltd
PO Box 92
Reading
RG3 4EF

A Brief Disc-ussion....

Roland Perry whets your appetite for the disc system and reveals some of those invaluable hints and tips which make transferring programs to it that little bit easier.



What you get

The DDI-1 disc system comes complete with interface and system disc, not forgetting the AMSDOS ROM located inside the disc interface. It can be operated either as a 'fast cassette' using extensions and re-directions of the built-in Locomotive BASIC or independently under the industry standard CP/M 2.2 operating system. These two alternative environments are designed to complement one another, rather than compete.

AMSDOS allows a powerful and flexible use of the disc system for loading and saving your BASIC programs written in the powerful built-in BASIC plus fast loading of disc versions of all the popular cassette games and tutorial/ business software. CP/M provides access to the wealth of 'serious' applications which have been considerably enhanced by the low-cost program suites including business accounts, word processing and database now available for the CPC464. (Not forgetting the bundled Dr. LOGO).

Compatibility between AMSDOS and CP/M

There is considerable compatibility between AMSDOS and CP/M because both use the same directory and file format and any information recorded in plain text (ASCII) is interchangeable between the two environments without a second thought. This means that you can use a CP/M word processor, often more powerful, to edit your Locomotive BASIC or DEVPAC programs. Each of these has the option to save their output in ASCII (',A' in BASIC, 'Q' in DEVPAC) and accept ASCII automatically.

Data files generated by Locomotive BASIC are in ASCII. In fact they 'look' very much like a set of DATA statements with each variable-length entry separated by a comma. These files can also be examined, generated or altered from within CP/M. Using the TYPE command from CP/M can be very instructive when glancing at ASCII files as it shows the exact structure visibly on the screen.

The DDI-1 User Manual examines the issues of file compatibilty and information transfer at some length. Note that any programs currently on tape or disc for use with the built-in

BASIC which are either protected BASIC, standard BASIC or Binary will make little sense to CP/M, although the standard (unprotected) BASIC could be LOADed and then SAVEd ',A'. Similarly the built-in BASIC cannot be expected to make much sense of non-ASCII CP/M files.

Compatibility between Cassette and AMSDOS.

In general the disc behaves like a 'fast cassette'. Programs load in a few seconds and data files can be interrogated and updated constantly throughout the running of a program. When dealing with program development, using DEVPAC or similar language tools, or even just playing the artist with Screen Designer, the increase in throughput when using a disc is quite startling. Most programs, written in BASIC or assembler plus firmware calls, are immediately portable, but there are a few wrinkles we have picked up that are worth mentioning.

Error messages for the disc system tend to be more complicated, and can require user input. It is prudent to reserve screen space for these messages. Certain AMSDOS messages are produced as a side effect of, for example, testing if a file exists on disc. The user should be made aware of which messages are to be expected, and what action, if any, to take.

When a CPC464 goes through the 'early morning' sequence it scans for external ROMS. When the DDI-1 is installed the AMSDOS ROM is discovered. This offers to the BASIC certain additional commands, for example `|DIR` which displays the disc directory. In addition, all the cassette functions are redirected to the disc.

Thus `CAT`, `RUN`, `WRITE #9` etc are performed using the disc. This re-direction can be modified later by issuing extension commands such as `|TAPE.OUT`, `|DISC.OUT`, `|TAPE.IN` and `|DISC.IN` which make the firmware switch between the disc and tape for output and input operations.

Because the jump block has already been re-directed once, by AMSDOS, in a special way, there is a particular scheme which has to be used if further re-direction is employed by the user program. This is described, along with many other firmware considerations, in Chapter 5 of the disc user manual.

The extended commands are available to the assembler programmer with the general rules published in SOFT158 The Concise Firmware Specification and their particular software interfaces documented in SOFT158A The Disc Firmware Manual. In particular it is important to pass the correct number of parameters to these commands, both in BASIC and in assembler, not forgetting the special case in assembler where there are no parameters (Send A=0).

Some of the CP/M BIOS facilities are available to the assembler programmer by extra extended commands not available to BASIC.

If you write assembler programs which are themselves `RUN` or pass control using
`MC_BOOT_PROGRAM`
then the early-morning initialisation of AMSDOS is one of the many things reset so that the new foreground program has a clean machine in which to operate.

The Disc Firmware Manual describes how to re-initialise AMSDOS in these circumstances if it is required at all, after the program has loaded.

As we have observed, there is compatibility between AMSDOS and CP/M files. This compatibility extends to filenames which must therefore conform, in particular, to the convention of having a maximum of eight characters plus three character extension. As every file on a disc has to have a name, and the concept of 'next file' has no meaning your program should refuse to accept (and obviously not prompt for) null filenames as is common on tape based applications.

A further implication of the CP/M compatibility is the detection of soft end-of-file whenever the &1A character is read. In certain circumstances the BASIC MERGE command will fail to operate as expected. Amsoft has a standard scheme to prevent this, which will be published in the next issue, along with various other recommended pieces of code to perform various utility actions such as detecting if there is a disc installed.

The AMSDOS ROM requires RAM as workspace and allocates itself &500 bytes. Another 4 bytes are used by the operating system. In BASIC this effectively lowers HIMEM, reducing the memory left for programs by 1280 bytes. Any BASIC program which assumes the value of HIMEM, loading for example user defined characters or assembler routines into fixed addresses is bound to be in difficulties.

The programmer should ensure that no such assumptions are made and that all such operations are effectively relocatable. Any other extension ROMS or RSX's (as published elsewhere in this issue) will also require RAM, further reducing the value of HIMEM. If it is unavoidable for binary programs to run overlapping the AMSDOS RAM then it is possible to load them at a lower address and have them move up store as their first action - as long as AMSDOS is no longer required!

Finally, there is a small piece of program that should be installed at the start of all BASIC programs that use data files or perform CHAIN or MERGE. This is just as valid for tape-only systems. The purpose is to permanently allocate the buffer used by the filing system, thereby avoiding various delays and side-effects as the dynamic buffer allocation interacts with the variable space and garbage collector. Being careful to avoid any interaction with `SYMBOL AFTER` commands use the following:

```
OPENOUT "DUMMY"
MEMORY HIMEM-1
CLOSEOUT
```

Compatibility between Amstrad CP/M and other CP/M's.

Our CP/M behaves as if it was running in 44.75k of RAM. This is less than the 64k offered by many up-market business machines. They do not use part of their RAM for a pixel screen nor do they generally have such a sophisticated firmware to support. Certain CP/M programs will therefore simply not fit, or be very 'uncomfortable'. Amstrad has

demonstrated, however, by its range of disc based CP/M software that it is perfectly possible to adapt very serious applications onto the CPC464. Innumerable utilities and languages will operate in the RAM size provided.

The interface to CP/M is very standard, and there is a RAM based BIOS jump block, although the BIOS itself is in ROM and the position of the BIOS jump block is determined from the size of CP/M loaded from disc. The normal facilities of the firmware are available, either through an entry point called `ENTER_ FIRMWARE` (which takes care of the various housekeeping necessary), or by invoking the control character commands described in Chapter 9 of the CPC 464 User Manual. Beware of control character commands which take parameters which are then intercepted either by the CP/M language environment or the BDOS. In particular many BASICs and the BDOS 'C=2' entry point expand the Tab character CHR$(9) to a number of spaces. This will upset the control character command sequence. The BDOS 'C=6' does not expand Tabs, but it can be difficult to avoid this problem from languages. An assembler subroutine appealing straight to the firmware is often the only solution.

Programs which require the second register set are catered for, but only restart 6 (&30) is free. Most programs avoid restarts because their use has always been in dispute by compatibility purists. The DDT supplied with the CP/M has been tailored to use restart 6 rather than the normal restart 7.

It is possible to add a $5^1/_4$" inch drive to a DDI-1 but there is little point in so doing. No extra capacity is gained, because the software is unaware that mechanism is a $5^1/_4$" rather than a 3". However, format is supported to allow software portability via the CP/M 86 single sided format of the IBM PC. Details of the mechanical arrangement are available in the Disc Firmware Manual, and the 51/4 inch mechanism has to be carefully chosen. to be suitable.

Software houses are already in a position to retail on the 3 inch format, so portability is not an issue. Some software houses are also prepared to undertake the custom installation procedure which most CP/M software requires.





Choosing a printer is a lot easier than choosing a computer.

THERE are dozens of quality printers from which to choose. With quality price tags of around £250.

The Brother M-1009, however, breaks all the rules.

Stays defiantly below the £200 barrier.

Though it has far more than its fair share of features, it maintains the extraordinarily low price of £199.95.

Travels at a steady fifty.

In the speed stakes, the M-1009 is certainly no slouch, being fully capable of up to 50 characters per second.

Providing bi-directional and logic seeking printing for normal characters and uni-directional printing for super and sub script and graphics.

Prints on any paper.

Being an impact printer, the M-1009 will print on virtually any paper, including letter headings, invoices and standard office stationery.

It will even print two copies together with your original.

A superb character recommendation.

In its price range, the M-1009 has a great deal more character than many printers.

96 no less, plus international type and graphic characters.

Reliability comes as standard.

Built to the same exacting standards as Brother's elite office printers, the Brother M-1009 already has faultless credentials for reliability.

Its 9 pin dot matrix head, for example, has an astonishing 20 million character service life.

One printer that doesn't block out the light.

Many home computers tend to be a little on the large side.

In contrast, the compact M-1009, at only 7 cm high, keeps a discreet profile.

Well designed, reliable – and conscientious.

The Brother M-1009.

brother®

The future at your fingertips.

DEPT P, BROTHER OFFICE EQUIPMENT DIVISION, JONES + BROTHER, SHEPLEY STREET, GUIDE BRIDGE, AUDENSHAW, MANCHESTER M34 5JD.
TEL: 061-330 6531 (10 LINES) 061-330 0111 (6 LINES) 061-330 3036 (4 LINES) TELEX: 669092
BROTHER INDUSTRIES LIMITED, NAGOYA, JAPAN.

—— AVAILABLE FROM ——

BOOTS, WILDINGS, SPECTRUM, JOHN MENZIES, MICRO MANAGEMENT, MAJOR DEPARTMENT STORES AND ALL GOOD COMPUTER SHOPS.

PASCAL's Blaze of Glory

In this article, we shall concentrate on just one of them, the popular and powerful language called Pascal. Our aim here is to show what makes PASCAL special and why programmers should seriously consider learning this language.

Pascal was invented by a Swiss gentleman called Niklaus Wirth (pronounced 'veert') at the Federal Institute of Technology in Zurich. It was devised primarily as a teaching langage because Wirth felt that the traditional languages - especially FORTRAN and BASIC - encouraged poor programming style. The chief difference between Pascal and the language most readers will be familiar with - BASIC, comes down to the vexed question of style.

But what is good style? Suffice it to say, for the time being, that good programs can be written in BASIC, but BASIC is a very forgiving language that allows and even encourages poor programming habits. These bad habits, once learnt, can be very hard to break. Pascal, on the other hand, is a strict language that insists on well structured programming. If the programs aren't properly thought through before they are written, they won't work.

Pascal is now the preferred teaching medium in almost all computer studies courses and the language is available on computers of all sizes, from home computers such as the CPC464 to monster main-frames. A whole generation of programmers today cut their teeth on Pascal and like it so much that they use it as a working language for writing commercial software. In this respect Pascal has had a similar history to BASIC, which also started life as a teaching language and then came into widespread general use. But there the similarities end. BASIC is an 'interactive' language that is usually interpreted, not compiled-see below. Pascal is a non - interactive language that is always compiled. It's the 'interactiveness' and interpreted nature of BASIC that makes it so different from Pascal, and that makes working in the two languages feel so different.

Compiled and Interpreted Languages

Most computer languages, especially those that run on main-frame computers, are what are known as compiled languages. What this means is that after the program has been written (this version of the program is called the source code) it is processed by another program called a compiler. The compiler produces a file of machine code instructions. This is the version of the program that is actually used by the computer when the program is 'run', and this version of the program is called the object code. Later we'll see how a program written in Pascal, a compiled language, actually gets from source code to object code.

Interpreted languages, of which BASIC is the best known example, work differently. The source code is written first, using the built-in editor supplied with all versions of BASIC. When the RUN command is typed in, a part of the program called an interpreter takes over and converts the source code, a line at a time, into object code. The object code is not stored as a separate file and the interpreter has to interpret (convert) each line of the source code to object code every time the program is run.

These differences may not seem very significant, but to a great extent they lie at the heart of what makes BASIC so easy to write programs in and what makes PASCAL comparatively more difficult.

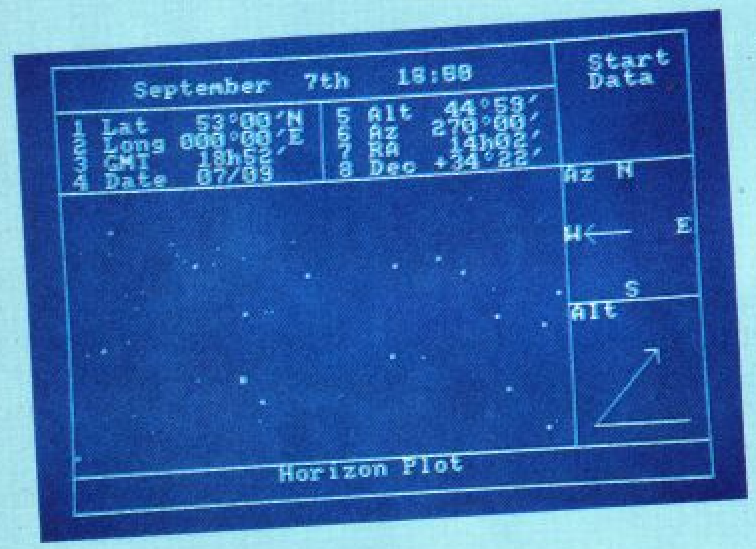
A Word or Two About Editors

The first difference in practice between the two languages is that BASIC is always supplied with a built-in editor but most versions of Pascal are not. When BASIC is loaded into the computer (it's already there waiting to run in the CPC464) it is in the 'command mode', which allows lines of code to be entered and edited. This editor is a part of the program that allows characters - letters, digits, punctuation marks and so on - to be entered on the screen, deleted or modified. People normally use the built-in editor to write their BASIC source code, athough it is equally possible to use a different editor to write the program - many wordprocessors will do it nicely.

The editors supplied with most versions of BASIC are 'line editors'. That means they allow you to process text a line at a time. A command such as `EDIT 100` will bring the line with that line number up on the screen and allow the programmer to modify it (by inserting extra characters, deleting characters etc.). The `RUN` command initiates the interpreter, which not only produces object code for the CPU to execute, but also doubles as a syntax checker. This means that each line of source code, before being converted into object code, is checked to see if it has the correct 'syntax'. If the line is not 'well formed' (i.e. does not conform to the rules of the language) this fact will be instantly communicated to the user by means of an error message displayed on the VDU screen. A line such as `120 FOR A=1 TO 10` is likely to result in a message such as `Syntax Error in 120`, if typed in wrongly. Other types of error, such as trying to READ a piece of data into an array that is too small will result in other more or less helpful messages such as `Subscript out of range in 150`. The line number in the error message instantly informs the programmer where to look and the mistake can be put right, there and then.

The best home micro now has the best training and applications software.

AMSOFT's Brainpower range from Triptych Publishing is simply superb. From star gazing with ***Star Watcher*** to understanding the principles of modern business planning with ***Entrepreneur***, each item in the Brainpower series provides an absorbing teaching program and an indispensable applications program to show you how to put what you've learned into practice.





Star Watcher will show you how to identify constellations and specific stars. It will explain the principles of astronomical navigation, and provide you with self assessment tests along the way.

Turn over the cassette, and you will be able to use the applications program to display the sky exactly as you would see it at any given time, from any given location on the earth's surface. The distinction between daylight, twilight and night is clearly visible. Amazing.

And if that's not enough for all you budding Patrick Moores, **Star Watcher** features a unique real time update mode, so you can set the program to display the sky and update the changes every minute. Even more amazing.

AMSTRAD CPC464

It's this kind of instant feedback that makes BASIC 'interactive', coupled with always having the built-in editor there ready to help with de-bugging. Contrast this process with a typical programming session in Pascal. (The Hisoft Pascal available for the CPC464 is actually a good deal easier to use in some ways than other Pascals, but the following general description is to give you a clearer overall picture of Pascal programming).

Programming in PASCAL

Pascal is usually supplied without an editor, though there are exceptions and Hisoft Pascal is one of them. On powering up the computer, the programmer's first task will be to load an editor. The kind of editor this will be depends in part on the type of computer system being used. On a moderately large home computer with disc drives, it is likely to be a full screen editor such as WordStar or The Final Word.

The program is written using the editor or wordprocessor as though it were an ordinary document. Suppose the program is one that adds all positive integers from 1 up to a specified limit. We have to give the program a name (that's one of the rules of Pascal) so we might decide to call it SUMNUM. Having written the source code as though it were an ordinary document using the editor or wordprocessor, we would save it as a file on disk or tape. The exact form of filename given to the file depends on the operating systems being used and the particular Pascal implementation. With Hisoft Pascal, SUMNUM will do fine as a filename. Under the CP/M disc operating system, a likely filename would be SUMNUM.PAS. The program we have just written is what programmers call the 'source code'. The computer (or rather the computer's Central Processing Unit) can't actually do anything with the program in this form; so before it can be 'run' or 'executed', it has to be converted by another program into binary code. This version of the program is called 'object code'.

Having saved the source code program under a suitable filename, the programmer exits from the wordprocessor program and returns to the computer's operating system (CP/M in the case of most disc-based computers). He then loads the Pascal compiler (a compiler is a special program that converts high level source code programs into machine language object code). The Pascal compiler, once it has loaded, then asks for the name of the source program to be compiled. The compiler then gets to work to turn the source code into a form the computer can execute directly.

The compilation process is fairly slow. Compilers usually have to go through the source code several times before they can produce executable object code, some as many as three to five times. The Hisoft compiler manages to do its job in a single pass, and is thus known as a one pass compiler.

Mind Your P's and Q's

As well as generating the object code file, the compiler also checks the source code for syntax errors. The rules for writing programs in Pascal are very strict and the likelyhood of small errors in the program cropping up is extremely high. Every time an error is encountered, a missing semi-colon, for example, or a missing END statement, the compiler will report the error with a screen error message. If an error is encountered during compilation, the programmer will have to exit from Pascal, call up the wordprocessor or editor, get the file containing the source code program, locate the error, correct it, save the corrected file again, exit from the wordprocessor, invoke the compiler and try compling the source code again. All very time consuming!

With multi-pass compilers it is always possible for an error to be picked up during the compiler's first pass and fixed. Having done that it is possible for still further errors to emerge during the second or third pass of the compiler. Again, these will have to be corrected before an object code file can be produced. If the entire compiling process takes place without any errors being reported, we end up with executable object code saved in a file and ready for instant execution.

Even at this stage it is possible for so called 'run time' errors to occur. This happens when the program has been formulated according to the rules but some logical flaw in the program allows unsuitable data to be entered and processed. The good news, however is that Pascal, by its very nature, forces programmers to write well structured programs in which errors of logic are far less likley than in programs written in 'unstructured' languages such as BASIC.

Not All Tears and Gnashing of Teeth

If programming in PASCAL demands more care and rigour, and if debugging is such a time consuming and frustrating activity, you might wonder why Pascal is now the world's second most popular programming language. The reason is that Pascal has strengths and power that make BASIC look like a baby-talk language in comparison. One of its strengths is that, because it is a compiled language, it runs very much faster than interpreted programs written in BASIC. This happens because the entire high level program (the source code) has been converted to machine code (the object code) before it is run. The original source code program is 'thrown away' as it were, and only the object code is used when the program is run.

With an interpreted language, lines of source code are converted to object one at a time as the program is run. This means that a copy of the original source code has to be kept and translated a line at a time every time the program is used. An analogy with 'human' translation goes like this: Imagine a Commander of a Roman garrison who periodically has to read the riot act to tribes of unruly Celts. The text, in Latin, runs along the following lines:

O hairy barbarians
If you do not resist from
raping our cattle and pillageing our women
Then I shall have no alternative but to raze your villages to the ground and decimate your ranks
On the other hand
If you would like to live in peace and prosperity
Then pay your tithes and taxes and
get the hell out of here

If the commander had a basic sort of mentality, he would translate this (as he read it out) a line at a time into his best but rather faltering Welsh. If, however, he was all for speed and efficiency, he would translate the whole thing into Welsh and write it down on a piece of parchment (keeping the Latin original out of harms way in the library in case it was needed again later). Then, any time he needed to tick off the natives, he could pull out the Welsh version and rattle off the whole thing in next to no time, thereby getting his message across that much quicker and possibly saving a lot of bloodshed.

The translation process from source to object code is always slow. With an interpreted language, this slow translation takes place every time the program is run. With a compiled language, the translation process (the compiling stage), happens only once. It may be slow, but it only needs to be done once and the resulting object code program will run much faster.

Exempli Gratia, Boyo

Let's illustrate the difference in speed made by compiling a program (rather than interpreting it) by writing a simple program in both Pascal and BASIC. The program below sums all the positive integers from 1 up to the TOP number (which is input from the keyboard when the program is run). First type in the BASIC program, type run and enter 10000. The program will add all the positive numbers from 1 to 10000 and give the sum on the screen. In BASIC this process takes 49 seconds (a very creditable performance, incidentally; in Microsoft BASIC running on a large Japanese desk-top business computer costing several thousand pounds, it took 69 seconds). If, however, you have the Hisoft Pascal compiler, type in the Pascal version of the program and try it with the same number. It executes more than four times as fast, taking just 11 seconds.

Speed. Who Needs It?

Speed may not be important to your programs. If you are writing a program to produce company pay-slips, a program that takes an hour and a half to run may be of no significance if it can be allowed to run overnight. If, on the other hand, you need a program in a 'time-is-money' situation, the time advantage of a good Pascal compiler could be worth its weight in gold. Hisoft's Pascal compiler overcomes several of the disadvantages of conventional compilers. It has, for example, its own built-in line editor, and that saves the need for jumping in and out of the operating system.

If the drift of this article has seemed biased against Pascal as a programming language, it is only because the time spent in coding, as against running, the program, is likely to be greater for a program written in Pascal than for one written in BASIC. BASIC allows sloppy and ill-conceived programming habits which make for easy-but-dirty programs. Pascal insists on more rigour; it may take more time at the coding stage, but you are almost guaranteed better structured programs that are less likely to contain errors of logic and that almost always run far faster than programs written in interpreted BASIC.

The choice is simple: easy-to-use, error prone BASIC, or slightly more demanding, elegant and secure Pascal. For really big programming projects, where well-constructed programs are a must, Pascal is definitely the language of choice. Large, commercial programs are almost never written in BASIC, and now you know why.

```
10 REM COMPUTES THE SUM OF ALL INTEGERS THRU TOP
20 INPUT "INPUT TOP NUMBER";TOP
30 NMB = 1
40 FOR X = 1 TO TOP
50 SUM = SUM + NUM
60 NUM = NUM + 1
70 NEXT X
80 PRINT "THE SUM OF NUMBERS 1 THRU ";TOP;" IS ";SUM
90 END
```

The SUMNUM program coded in BASIC

```
PROGRAM SUMNUM;
(*COMPUTES THE SUM OF ALL INTEGERS THRU TOP*)
VAR
   NUM, SUM, TOP : REAL;
BEGIN
 (* READ TOP NUMBER *)
 WRITE ('INPUT TOP NUMBER');
 READ (TOP);
 (* INITIALISE VARIABLES *)
 SUM := 0;
 NUM := 1;
 (* COMPUTE THE SUM OF NUMBERS 1 THRU TOP *)
 WHILE NUM <= TOP DO
   BEGIN
    SUM := SUM + NUM;
    NUM := NUM + 1;
   END (* WHILE *)
  (* PRINT SUM *)
  WRITE ('THE SUM OF NUMBERS 1 THRU ',TOP,' IS',SUM)
 END
```

SUMNUM coded in Pascal. Enter the TOP number when prompted and the program will sum all integers from 1 to that number. Note the difference in speed between BASIC and Pascal.

Trench Attack!



Trench is a continuation of David Robinson's trench program in the last issue of the user magazine. You fly down the trench towards the vent at its end, blasting the baddies for all you are worth and prevent Kill-O-Zap plasma bolts from flying up the tailpipe of your X-wing fighter.

Your dual 5000 Megahurt Maxi-Slorta Hyper Blasters only fire forwards (in perspective they appear to head for the centre of the screen) whereas the enemies can fire in any direction, but you have the edge on range.

At the end of the trench (approximately 6 minutes flying time) you will encounter the vent. This must be hit dead centre to cause the desired explosion. Don't forget to fly over the wall at the end of the trench or the CPC464 will demonstrate what a wonderful explosion effect it can produce!

These wonderful effects are also shown if you fly into a wall,get hit by enemy fire or fly too high when you will be rapidly picked off by fire from the surface (except at the end of the trench).

It is very difficult to blast an alien from directly above, below or to the sides due to their shielding but they eventually succumb to fire from even a slight angle.

You know how far down the trench you have gone as this is indicated by a bar in the top right corner of the screen.

Points are scored for blasting aliens, hitting the vent and (hopefully) your triumphant exit from the trench without one of those explosion effects.

Points are deducted for firing weapons around like Wyatt Earp in the O.K. Corral.

Movement and blasting is done with the cursor cluster and space bar or joystick.

The closer an alien is to you the more points you get for blasting him. The problem is that he is more likely to blast you.

All you have to do now is remember to aviod crashing into the walls and may the force be with you!

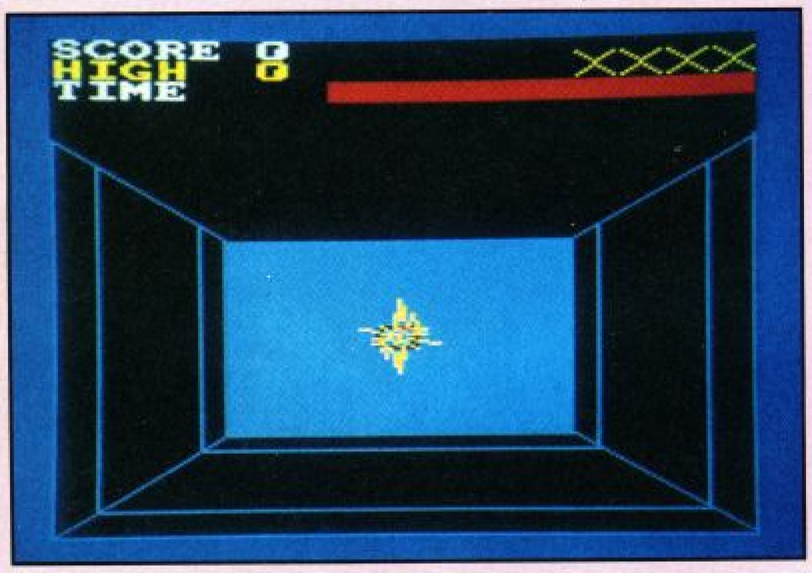
A brief description of various parts of the program:

110 -	Set keys for a fast repeat rate.
110 - 220	MICR style character set.
230 - 240	Calls up instructions, title page and sets graphics plot mode to normal.
250	Creates wierd noise and flashes title page.
260 - 300	Sets up the screen for an alien zapping session.
310 - 340	Initializes event timers for moving trench & end of game. Also sets graphics plot mode and launches your first fighter. If you want a shorter or longer game make the figure 1500 smaller or larger.



350	Check for joystick movement, key presses, furious pounding of keyboard etc.
360	Sees if you have run out of time yet.
370	Moves you again so that you go faster than the enemy.
400 - 420	Draws the X-wing fighter. PLOTRs prevent unsightly holes in fuselage.
430 - 470	This bit draws the trench.
480 - 530	Tidies up ragged edges from previous trench drawing.
540 - 560	Cycles inks 1,2 & 3 so trench appears to move.
580 - 630	Read keyboard or joystick so you appear to move.
640	Looks to see if you fired the blasters.
650 - 700	Various ways of blowing yourself up and getting reincarnated.

710 - 740	Shoot bang fire!
750 - 850	Various jiggery pokery to deal with the odd occasion when you actually hit anything.
860	Display score (if any).
880 - 910	Draw the alien (Boo! Hiss!).
920 - 1030	Decide path of alien.
1040 - 1070	Put him on that path.
1080 - 1160	The alien strikes back and occasionally hits you (so don't get too cocky).
1170 - 1250	Aforementioned neat explosion effects (red for him, yellow for you).
1260 - 1320	Oh dear, all this fun is over. See if promising young starfighter wants another crack at it.
1230	As parting gift give the user readable text and a useable keyboard.
1350 - 1500	Stop any timers and display instructions for those who have not read the article properly.



1510 - 1540	Updates that curious red and yellow progress indicator and initiates wall routine when time is up.
1550 - 1610	Deals with joystck.
1620 - 1740	Print appropriately sized wall and vent without obliterating spacecraft.
1750 - 1780	This is when you hit 'em where it hurts.
1790 - 1850	and scraper without becoming an ex-fighter pilot.

```
100 SPEED KEY 5,2
110 '*** Redefine the Numbers
120 SYMBOL AFTER 32
130 SYMBOL 48,&7C,&C6,&C6,&CE,&CE,&CE,&7
    C
140 SYMBOL 49,&18,&18,&18,&38,&38,&38,&3
    8
150 SYMBOL 50,&7E,&66,&6,&7E,&60,&6E,&7E
160 SYMBOL 51,&7E,&66,&6,&3E,&E,&6E,&7E
170 SYMBOL 52,&CC,&CC,&DC,&FE,&1C,&1C,&1
    C
180 SYMBOL 53,&7E,&66,&60,&7E,&E,&6E,&7E
190 SYMBOL 54,&7E,&66,&60,&7E,&6E,&6E,&7
    E
200 SYMBOL 55,&7E,&66,&C,&C,&1C,&1C,&1C
210 SYMBOL 56,&3E,&36,&36,&7E,&6E,&6E,&7
    E
220 SYMBOL 57,&7E,&66,&6E,&7E,&E,&E,&E
230 wall=0:GOSUB 1340:INK 0,0:BORDER 2:I
    NK 1,11:PEN 1:PRINT CHR$(23)CHR$(0);
    :life=4
240 PAPER 0:MODE 0:LOCATE 8,10:PRINT"TRE
    NCH":PRINT:PRINT" "CHR$(164)" Suzzy
    Billingham"
250 SOUND 1,499,500:SOUND 2,500,500:FOR
    j=1 TO 16:INK 1,j:FOR r=1 TO 99:NEXT
    :NEXT
260 PAPER 0:CLS:ENV 3,5,10,20:y=380:FOR
    x=500 TO 620 STEP 40:GOSUB 400:NEXT
270 GOSUB 430:ENT 1,239,20,1:ENV 2,7,-2,
    2
280 tv=1/12:DEF FNt(x)=INT(x*tv+0.5)*12:
    d=0:prg=0:LOCATE 1,3:PEN 14:PRINT"TI
    ME     ";:PAPER 4:PRINT"              "
    ;:PAPER 0
290 FOR i=1 TO 3:INK i,13:INK i+4,24:INK
     i+7,6:NEXT:INK 4,24:INK 11,6
300 inv=0:INK 14,26:PEN 14:LOCATE 1,1:PR
    INT"SCORE 0":PEN 4:PRINT"HIGH ";STR$
    (hi);
310 EVERY 5,1 GOSUB 540
320 EVERY 1500,2 GOSUB 1510
330 PRINT CHR$(23)CHR$(1);:xp=320:yp=160
    :xm=320:ym=180
340 x=xp:y=yp:GOSUB 400
350 GOSUB 570
360 IF wall THEN GOSUB 1620 ELSE GOSUB 9
    20
370 GOSUB 570
380 GOTO 350
390 '
400 '*** Draw Spacecraft ***
410 MOVE x-16,y+8:DRAWR 16,-8,4:PLOTR 0,
    0:DRAWR 16,8,4
420 MOVE x-16,y-10:DRAWR 16,8,4:PLOTR 0,
    0:DRAWR 16,-8,4:RETURN
430 '*** Create Trench ***
440 j=0:FOR i= 0 TO 23:j=j+0.2:x=2mj:x=I
    NT(x*10)/10
450 ik=i MOD 3+1:MOVE x*10+320,160+x*5:D
    RAWR 0,-x*10,ik
460 DRAW 320-x*10,160-x*5,ik:DRAWR 0,x*1
    0,ik
470 NEXT
480 FOR i=320 TO 160 STEP -2
490 MOVE i*2,i:DRAWR (160-i)*4,0,0
500 NEXT
510 MOVE 0,320:DRAW 0,0,3:DRAW 639,0,3:D
    RAW 639,320,3
520 INK 15,14:MOVE 0,320:DRAW 640,0,15:M
```

NEW

ON

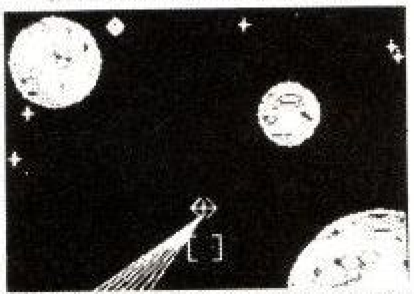
AMSTRAD

BY POPULAR DEMAND!!!

5 EXCITING HIGH QUALITY GAMES FROM DATACOM AT ONLY £5.95 each!

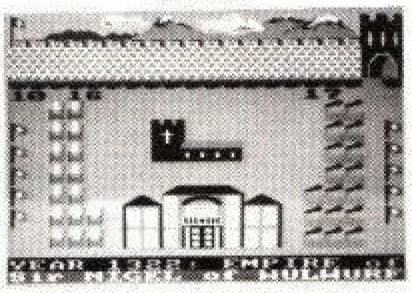
* SPECIAL OFFER * ANY 3 FOR £14.95 ALL 5 FOR £22.95

INTERPLANETARY TRADER



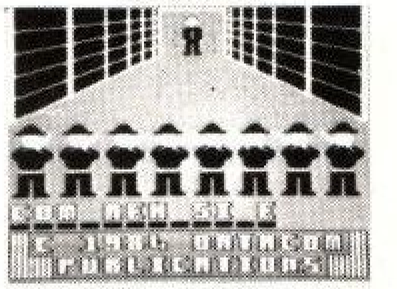
Space adventure at its best! Face the perils of space pirates, meteors, and other hazards as you penetrate the depths of space in your quest to become a GALACTIC MEGA-BILLIONAIRE! A multilevel, feature packed game of strategy.

EMPIRE



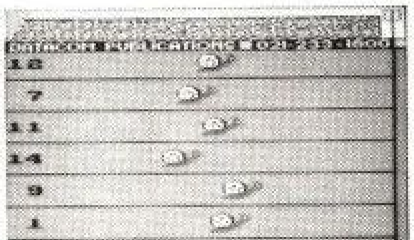
From a Knight to a King! Can you build up your empire in the face of stiff opposition? Will you keep subjects happy and balance the economy. This is an exciting multilevel game of strategy for up to 4 players. Includes a save game facility.

EXECUTION



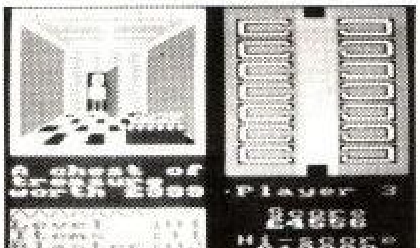
You need nerves of steel to play EXECUTION. Standing before the firing squad you realise this is your last chance to receive a pardon! but time is fast running out. A multilevel educational game with 1000+ word vocabulary.

SNAIL PACE



Thoroughbred racing snails thrash it out on the race course! Will you leave the course a rich man? This superb complex program features computer quoted odds, form guide, and magnificent graphics that will ensure hours of excitement.

WUMPUS MANSION



Only a CRAZY person would venture near WUMPUS MANSION. Tempted by hoards of treasure you raid the mansion and leave through a maze of tunnels. Will you outwit the dreaded WUMPI, avoid the timebombs and trapdoors? Compelling entertainment!

Please send Cheques/P.O to: DATACOM PUBLICATIONS, 407F Hockley Centre, Birmingham B18 6NF.

Name ....

Address ....

....

....

Tel No. ....

Please find enclosed my cheque/P.O. for £ ____________

	QTY	PRICE
INTERPLANETARY TRADER		
EXECUTION		
WUMPUS MANSION		
SNAIL PACE		
EMPIRE		
Please rush me copies of the above **TOTAL**		

DISTRIBUTORS/DEALERS (UK and Overseas)

Ring: DAVID READING on 021-233 1800 for our excellent terms.

```
OVE 0,0:DRAW 640,320
530 RETURN
540 '*** Cycle the Trench ***
550 INK tren+1,0:tren=(tren+1) MOD 3
560 INK tren+1,14:RETURN
570 '*** Move Player Spacecraft ***
580 ox=xp:oy=yp:j=JOY(0):IF j>0 THEN 1550
590 IF INKEY(1)+INKEY(2)+INKEY(0)+INKEY(8)+INKEY(47)=-5 THEN RETURN
600 IF NOT INKEY(1) THEN xp=xp+12:IF xp>=625 THEN 650
610 IF NOT INKEY(8) THEN xp=xp-12:IF xp<=15 THEN 650
620 IF NOT INKEY(2) THEN yp=yp-12:IF yp<=10 THEN 650
630 IF NOT INKEY(0) THEN yp=yp+12:IF yp>=305 THEN 650
640 x=ox:y=oy:GOSUB 400:x=xp:y=yp:GOSUB 400:IF NOT (INKEY(47) AND INKEY(76)) THEN 710 ELSE RETURN
650 IF yp<305 THEN 670 ELSE IF wall THEN 1790 ELSE MOVE 20,300:DRAW ox,oy,14:DRAW 620,300,14
660 SOUND 1,200,10,3,2,1:DRAW ox,oy,14:DRAW 20,300,14
670 SOUND 2,600,60,4,3,0,8:x=ox:y=oy:GOSUB 1170:yp=160:xp=320:GOSUB 1170
680 GOSUB 400:x=xp:y=yp:GOSUB 400
690 life=life-1:IF life<0 THEN 1260 ELSE x=620-life*40:y=380:GOSUB 400
700 RETURN
710 '*** Fire The Hyper-Lasers ***
720 IF wall THEN 1750
730 MOVE xp-10,yp:DRAW 320,160,4:DRAW xp+10,yp,4:SOUND 1,60,0,3,2,1:pl=pl-5
740 MOVE xp-10,yp:DRAW 320,160,4:DRAW xp+10,yp,4
750 hit=0:IF SGN(320-xm)<>SGN(xm-xp)THEN RETURN
760 IF SGN(160-ym)<>SGN(ym-yp)THEN RETURN
770 mx=xm-320:my=ym-160:px=xp-320:py=yp-160
780 IF my=0 THEN gm=1E+09
790 IF py=0 THEN gp=1E+09
800 IF my=0 OR py=0 THEN 820
810 gm=mx/my:gp=px/py
820 df=MAX(ABS(gm),ABS(gp))*0.15
830 IF ABS(gm-gp)>df THEN RETURN
840 SOUND 2,800,50,4,3,0,8
850 x=xm:y=ym:GOSUB 1220:GOSUB 880:ym=160:xm=320:d=0:GOSUB 1220
860 inv=inv+INT(pl):PEN 14:DI:LOCATE 7,1:PRINT STR$(inv);:EI
870 RETURN
880 '*** Draw The Alien ***
890 p=8*d:q=16*d:MOVE x-p,y+p*0.5
900 DRAWR q,-p,8:DRAWR 0,p,8:DRAWR -q,-p,8:DRAWR 0,p,8
910 RETURN
920 '*** Move Alien Routine ***
930 IF d=0 AND RND>0.2 THEN RETURN
940 IF d<>0 THEN 1040
950 xf=INT(RND*640):yf=INT(RND*160):s=0:d=0.25:pl=101:xm=xf:ym=yf
960 IF (yf>148 AND yf<172) OR (xf>308 AND xf<332) THEN 950
970 ON INT(RND*4+1) GOSUB 980,990,1000,1010:GOTO 1020
980 xf=0:RETURN
990 xf=640:RETURN
1000 yf=0:RETURN
1010 yf=320:RETURN
1020 xc=(xf-320)/64:yc=(yf-160)/64
1030 RETURN
1040 s=s+2:pl=pl+0.5:ox=xm:oy=ym:xm=FNt(xc*s+320)
1050 ym=FNt(yc*s+160):x=ox:y=oy:GOSUB 880
1060 IF s>64 THEN d=0:RETURN
1070 x=xm:y=ym:d=d+0.05:GOSUB 880
1080 '*** Return Fire Routine ***
1090 IF s<SQR(RND)*32 THEN RETURN
1100 IF SQR((xp-xm)m2+(ym-yp)m2)>s*2 THEN RETURN
1110 IF RND>0.6 THEN RETURN
1120 MOVE xm,ym:DRAW xp,yp,8
1130 SOUND 4,100,10,3,2,1,2
1140 MOVE xm,ym:DRAW xp,yp,8
1150 IF RND>0.85 THEN ox=xp:oy=yp:SOUND 2,300,18,3,3,0,1:GOTO 670
1160 RETURN
1170 '*** Explosion Effect & Sound Routine ***
1180 ORIGIN x,y:c=3:GOSUB 1200
1190 c=6:GOSUB 1200:ORIGIN 0,0:RETURN
1200 FOR g=0 TO 2+c:MOVE g*c,(5-g)*c:DRAW-g*c,(g-5)*c,((c MOD 2)+1)*4
1210 MOVE(5-g)*c,g*c*0.5:DRAW(g-5)*c,-g*c*0.5,((c MOD 2)+1)*4:NEXT:RETURN
1220 ORIGIN x,y:c=3:GOSUB 1240
1230 c=6:GOSUB 1240:ORIGIN 0,0:RETURN
1240 FOR g=0 TO 2+c:MOVE g*c,(5-g)*c:DRAW-g*c,(g-5)*c,(((c+1) MOD 2)+1)*4
1250 MOVE(5-g)*c,g*c*0.5:DRAW(g-5)*c,-g*c*0.5,(((c+1) MOD 2)+1)*4:NEXT:RETURN
1260 '*** End of Game ***
1270 wp=0:INK 13,18,7:PEN 13:LOCATE 7,10:PRINT"GAME OVER"
1280 PEN 12:INK 12,13:LOCATE 4,12:PRINT"Another Game ?"
1290 LOCATE 9,13:PRINT"[y/n]"
1300 a$=LOWER$(INKEY$):IF a$<>"y" AND a$<>"n" THEN 1300
1310 IF hi<inv THEN hi=inv
1320 IF a$="y" THEN EI:GOTO 230
```

HISOFT

PASCAL

A powerful and virtually full implementation of Standard Pascal. A valuable educational and development tool in an incredibly small size (19K) for such a complete language compiler, compiles directly to Z80 code which executes very quickly, typically at least 40 times faster than the BASIC equivalent. INTEGERs, REALs, CHARs, ARRAYs, SETs, RECORDs, POINTERs, IF . . . THEN . . . ELSE, CASE . . . OF, WHILE . . . DO, REPEAT . . . UNTIL, FOR . . . DO, fully recursive procedures and functions, value and variable parameters etc. etc. So much that you will not believe it. You MUST write for details.

DEVPAC

An excellent, fast (4000 lines per minute) assembler coupled with a powerful disassembler/debugger. So many features that we cannot possibly do the package justice here – conditional assembly, assembly from tape, screen and line editing, full arithmetic, front panel debugger with disassembler, single step, multiple breakpoints, modify, list and move memory etc. Everything you need AND fully relocatable so that it works anywhere within the memory map.

HISOFT

180 High Street North
Dunstable, Beds. LU6 1AT
Tel: (0582) 696421

HISOFT

presents

FONT 464

for the

AMSTRAD CPC 464

FONT 464 *is a sophisticated and powerful font designer especially developed for the CPC 464 microcomputer, allowing you to design and print your own character fonts and graphic symbols in a friendly, menu-driven package.*

FONT 464 *allows you to create a new design or amend an existing one using such features as set, reset, invert, relect, rotate, inverse and even scrolling and animation!*
Load and save character sets from/to tape or disc, use the new character(s) from within BASIC, output the character(s) to an Epson or DMP1 printer, design your own animated graphics – all this and more with FONT 464!

FONT 464 *is supplied with five interesting and amusing character sets for you to experiment with and includes a full screen copy routine (written as an RSX) for both Epson and DMP1 printers.*

** All this power for: £7.95 inclusive **

Order FONT 464 directly from Hisoft or write for more details.

Please write to Hisoft for more details of Pascal and Devpac on the CPC 464 or contact Amsoft with your order.

New Serious Software for the AMSTRAD CPC 464 from Gemini @ £19.95

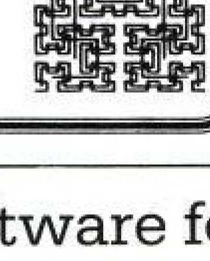
Database C.1801

Stores, retrieves and manipulates data with up to 20 fields, each with 160 characters – to a maximum of 1000 characters per card... up to 999 records available on user definable card layouts in 40 or 80 column mode... sorts ('Heap method'), alphabetically or numerically on subsets of data in ANY field... uses machine code routines... superbly documented with simple menu driven operation... integrates with Report Generator program to produce standard mail-merged form letters and labels.

Report Generator C.1820

Requires Database C.1801 and allows you to present and print data in user defined formats so as to :-

1. Re-arrange your record layout to meet specific printed requirements.
2. Create sets of mailing list labels from database records, up to three across your printer.
3. Create documents inserting merged fields into text, just like the expensive word-processors!
4. Design field summaries for columnar presentation of data, and total any given column.
5. Create any number of standard documents from one database file.



Home Accounts C.1807

The ultimate home accounting program featuring up to 30 expenditure allocations, computer aided budget design... bar charts of budget/actual income/expenditure... automatic warning if bank charges likely... varied printer routines... and many other useful features. Gemini, whose board of Directors includes 3 Chartered Accountants, produce some of the finest accounting software ever written, and ALL their experience and knowledge have been used to ensure that their Amstrad Home Accounts program will quickly establish itself as "the best in the business"!

Market leaders for serious software for the BBC Micro, CBM 64 and Spectrum, Gemini's vast experience has been used to produce three brilliant programs for the Amstrad, all designed to use its outstanding features, to the maximum.

Gemini

Gemini Marketing Ltd.
18a Littleham Road, Exmouth,
Devon EX8 2QG.
Tel. (0395) 265165

Available from: Larger branches of Boots and W.H.Smith and most good software dealers... or phone Gemini for prompt despatch quoting your Access, Visa, or American Express card number. Telephone (0395) 265165. Alternatively make your cheque or Postal Order payable to Gemini Marketing Ltd., and send to Gemini Marketing Ltd., 18a Littleham Road, Exmouth, Devon EX8 2QG.

Please send urgently C.1801/1807/1820 @ £19.95 each

Prices listed include VAT and Post and Packing (UK only).

Name ....

Address ....

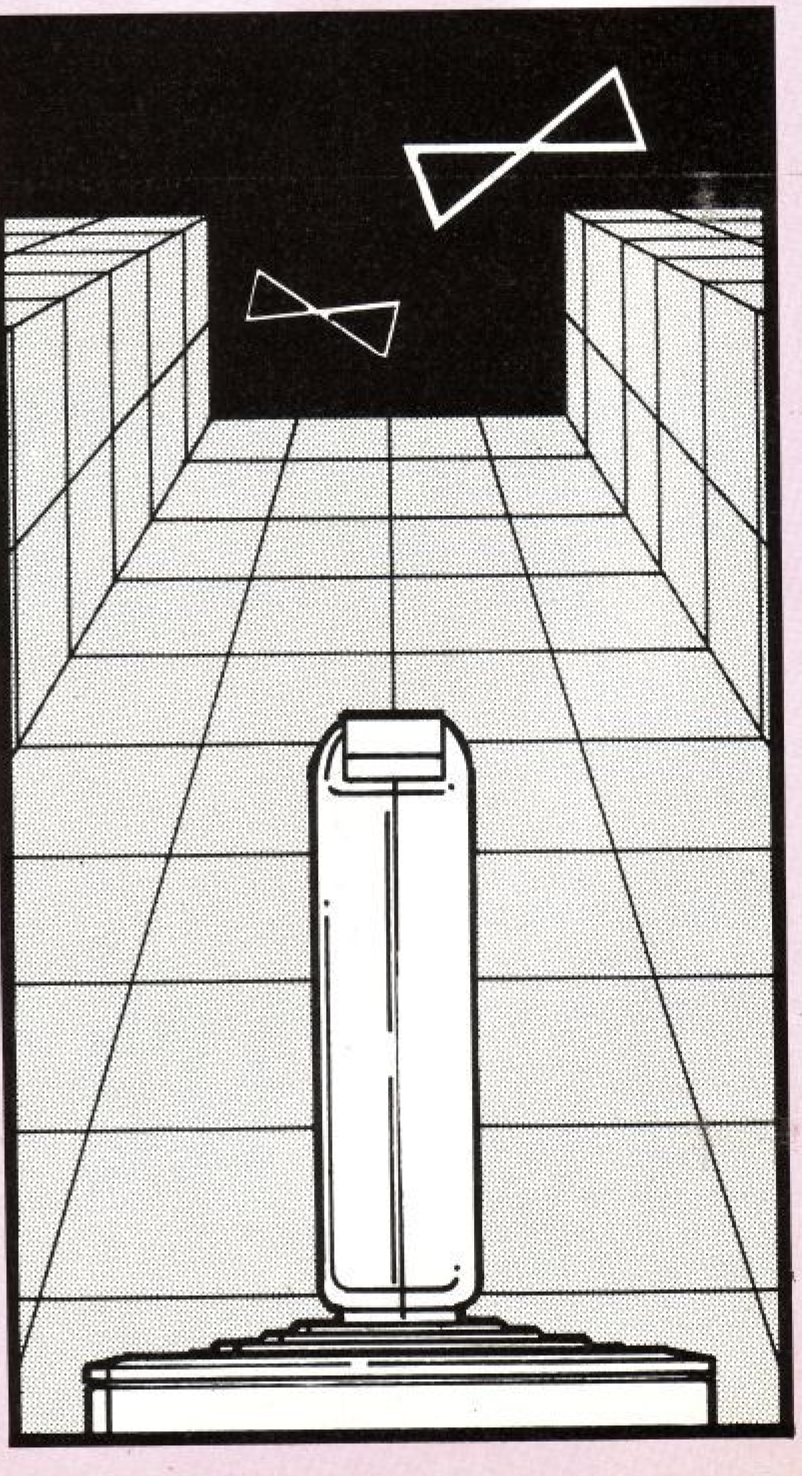
Signature ....

Credit Card No. Access/Visa/American Express (delete) ....

Please send further information concerning ....

AMU

```
1330 INK 0,0:PEN 1:PAPER 0:MODE 2:INK 1,
     26:SPEED KEY 20,2:STOP
1340 '*** Instructions ***
1350 i=REMAIN (1):MODE 1:PAPER 0:INK 0,0
     :PEN 1:INK 1,18
1360 PRINT"You are piloting a small X-at
     tack craft"
1370 PRINT"down a trench towards a vent
     at its end."
1380 PRINT"Your armament consists of for
     ward-firing";
1390 PRINT"dual 5000 Megahurt Maxi-Slort
     a Blasters which fire down the tren
     ch.":PRINT
1400 PRINT"The enemy has a shorter range
      version   which is, unfortunately,
      directable."
1410 PRINT:PRINT"Fly too high, however,
     and you'll be    picked off by surf
     ace fire."
1420 PRINT:INK 2,24:PEN 2:PRINT:PRINT TA
     B(12);"Keyboard Movement":PRINT
1430 PRINT TAB(19);"↑"
1440 PRINT TAB(19);"up"
1450 PRINT SPC(12);CHR$(242);" left righ
     t ";CHR$(243)
1460 PRINT TAB(18);"down": PRINT TAB(19)
     ;CHR$(241)
1470 PRINT:PRINT:PRINT"    [SPACE] or [FI
     RE] to fire blasters"
1480 PRINT TAB(10);"and also to continue
     ."
1490 IF JOY(0)<>0 THEN 1500 ELSE IF INKE
     Y$<>" "THEN 1490
1500 RETURN
1510 '*** Print Progress Down Trench ***
1520 LOCATE 9+prg,3:PAPER 8:PRINT" ";:PA
     PER 0
1530 prg=prg+1:IF prg<12 THEN RETURN
1540 wall=-1:RETURN
1550 '*** Joystick Movement ***
1560 IF (j AND 8)=8 THEN xp=xp+12
1570 IF (j AND 4)=4 THEN xp=xp-12
1580 IF (j AND 2)=2 THEN yp=yp-12
1590 IF (j AND 1)=1 THEN yp=yp+12
1600 IF xp>624 OR xp<16 OR yp>304 OR yp<
     11 THEN 650
1610 GOTO 640
1620 '*** Wall Routine ***
1630 IF wp>0 THEN 1670 ELSE i=REMAIN (2)
1640 wp=1:INK 11,24:PAPER #2,15:PEN #2,0
1650 x=xm:y=ym:GOSUB 880
1660 pl=1000
1670 x=xp:y=yp:GOSUB 400:xm=0:ym=0
1680 WINDOW #2,10-wp,11+wp,16+wp,15-wp
1690 CLS#2:LOCATE #2,wp+1,wp+1:PRINT#2,C
     HR$(150);CHR$(156)
1700 LOCATE#2,wp+1,wp+2:PRINT#2,CHR$(147
     );CHR$(153);
1710 GOSUB 400
1720 IF al=0 THEN al=1 ELSE al=0
1730 wp=wp+al:IF wp<10 THEN RETURN
1740 x=xp:y=yp:SOUND 2,600,50,4,3,0,8:GO
     SUB 1170:GOTO 1260
1750 '*** Fire at The Vent ***
1760 SOUND 1,60,0,3,2,1:IF hit OR xp<>32
     0 OR yp<>160 THEN RETURN
1770 SOUND 4,0,25,7,0,0,5:hit=-1
1780 x=320:y=160:GOSUB 1170:GOSUB 1170:G
     OTO 860
1790 '*** Triumphant Exit From Trench
1800 x=ox:y=oy:GOSUB 400:y=oy+28:GOSUB 4
     00
1810 up=-1:WINDOW #2,1,20,6,25:PAPER #2,
     0
1820 PLOT 0,320,15:PLOT 639,320,3
1830 FOR i=1 TO 21:PRINT#2,CHR$(11);:NEX
     T
1840 pl=250:GOSUB 860
1850 GOTO 1260
```



Fill of Machine Code

This program contains a general purpose routine which will fill any enclosed area of the screen in any chosen colour except the current paper colour. It will work in all screen modes and its operation can be aborted at any time before the area is completely filled.

The routine operates by scanning left and right from the starting point, filling in any unpainted pixels, until it meets a vertical boundary. It then stores its current values on a stack and moves up a line from where it calls itself, recursively, and repeats the process until it meets a horizontal boundary. When this happens the program retraces its steps - by retrieving previous values from the stack and continuing from where they left off -moving downwards if necessary, until all stored values have been removed from the stack and all pixels within the boundary have been filled.

This process, however, has the disadvantage of altering some of the values on the stack as it proceeds so that a complete record of all operations is no longer available.

The effect of this is that it is not easy to reverse the process if anything goes wrong; for instance, if the shape to be filled has a small gap in it then the ink will 'escape' and proceed to fill in the whole screen. For this reason a second stack has been used to store such information as is necessary to enable the process to be aborted at any time and the partly filled area to be returned to normal. This use of two stacks, requires a large amount of memory for storage and in theory this could amount to more than the entire screen itself takes up. In practice though, two stacks of 2K each should be sufficient for all but the most intricate and elaborate shapes.

To see the routine in action type in the BASIC program and run it. If you have made a mistake in entering the data then the program will stop and inform you of the offending line which can then be corrected and the program re-run. Note that you must enter the checksum at the end of each DATA statement.

If all is well the screen will clear to mode 0 and an irregular shape will appear on the screen inside a large square. The program will then proceed to fill the area around the shape and when this is completed it will fill the shape itself, but in a different colour. After a short delay the whole process will be repeated using modes 1 and 2 and will then start all over again with mode 0. Note that in the highest resolution mode both colours will be the same since in this mode only two colours can be present on the screen. Pressing any key will abort the current fill stage and 'unfill' the painted area before proceeding to the next stage.

When any section is completely filled there is ample time, especially in the higher resolution modes, to decide whether or not to abort. To stop the program completely press **[ESC]** twice.

The machine code routine can be stored anywhere in memory by changing the value of start in line 70. If you relocate the program in this way and find that it does not function correctly and if it worked before you relocated it then you have almost certainly missed out a '@' somewhere in the DATA statements. Note that when relocating the code you must leave at least 4K for the stacks and about 450 bytes for the routine itself so it should not be loaded above about &9900 or 39168 decimal.

The demonstration program can be removed by deleting lines 310 to 440 and 480 to 520 and the remaining lines can be used as part of a larger program. Note that your program must set X and Y to a point inside the shape being filled before calling the routine. Also the required graphics ink must be previously selected by PLOT a,b,n where a,b is an imaginary point off screen - such as 800,800 - and n is the required ink number.

```
10 REM ***********************************
   *************************************
20 REM ** AMSTRAD CPC464 User Magazine :
    programs from Issue number 4 *******
30 REM *Basic loader and demonstration p
   rogram for machine code FILL routine*
40 REM ***********************************
   *************************************
50 BORDER 13:INK 0,13:INK 1,0:INK 3,24
60 DEF FN hex(h$)=VAL("&"+LEFT$(h$,2))
70 start=30000:REM Change this value to
   relocate routine at a different addre
   ss
80 addr=start:ln=510
90 MEMORY start-1
100 REM Main loop to poke in values from
     data statements
110 total=0
120 READ h$:h$=UPPER$(h$)
130 IF h$="END" THEN 280
140 IF ASC(h$)<>ASC("@") THEN 210
150 h$=RIGHT$(h$,LEN(h$)-1)
160 lb=FN hex(h$):total=total+lb:h$=RIGH
    T$(h$,LEN(h$)-2):IF h$="" THEN 250
170 hb=FN hex(h$):total=total+hb:h$=RIGH
    T$(h$,LEN(h$)-2)
180 w=256*hb+lb-30000+start
190 POKE addr,w-256*INT(w/256):addr=addr
    +1
200 POKE addr,INT(w/256):GOTO 230
210 b=FN hex(h$):total=total+b:h$=RIGHT$
    (h$,LEN(h$)-2)
220 POKE addr,b
230 addr=addr+1:IF h$<>"" THEN 140
240 READ checksum:IF checksum=total THEN
     ln=ln+10:GOTO 110
```

```
250 PRINT"ERROR IN LINE";ln
260 END
270 REM Demonstration program to draw an
    d fill shape in different modes
280 xa=&76DE-30000+start:ya=&76E0-30000+
    start
290 FOR i=0 TO 2
300 MODE i
310 PLOT 220,50,3
320 DRAWR 200,0:DRAWR 0,200
330 DRAWR -200,0:DRAWR 0,-200
340 PLOT 250,100,3
350 DRAWR 20,120:DRAWR 40,10
360 DRAWR 20,-40:DRAWR 10,20
370 DRAWR 60,-60:DRAWR -80,-20
380 DRAWR -20,40:DRAWR -20,-40
390 DRAWR 80,-40:DRAWR -110,10
400 PLOT 800,800,1
410 x=240:y=150
420 FOR j=1 TO 2
430 POKE xa,x-256*INT(x/256):POKE xa+1,I
    NT(x/256)
440 POKE ya,y-256*INT(y/256):POKE ya+1,I
    NT(y/256)
450 CALL start
460 FOR k=1 TO 700:NEXT
470 x=280:y=150:PLOT 800,800,3
480 NEXT j
490 NEXT i
500 GOTO 290
510 REM Hex values for machine code FILL
     routine
520 REM Note that a '@' indicates a valu
    e to be changed during relocation.
530 REM The value at the end of each lin
    e is the sum of all entries on that
    line
540 DATA CDE1BB32@EC7647CDE7BB32@ED76B8C
    8CD@6876C0ED73@EE76CD11BC4704,4316
550 DATA 3E08CB3F10FC32@DC7621@F27E22@F0
    7621@F68622@F47E21000022@E276,3013
560 DATA 22@E47622@E676CD09BBDA@9076ED5B
    @F07621@E776010A00EDB8,3143
570 DATA ED53@F076CD@6876204FCD@E8752A@E
    27622@E6762A@E67622@DE76,3302
580 DATA 2A@F0762323235E23561313ED53@E07
    6CD@6D752A@E67622@DE762A@F076,3015
590 DATA 2323235E23561B1BED53@E076CD@6D7
    52A@E676ED4B@DC760922@E676,2898
600 DATA ED4B@E476A7ED4238BC28BA11@DE762
    A@F07623010A00EDB02B22@F076C9,3450
610 DATA 2A@DC7622@EA76CD@34762A@E876ED4
    B@DC76A7ED4222@E476EB2A@F47E,3680
620 DATA 722B73E5210000ED4B@DC76A7ED4222
    @EA76CD@34762A@E876ED4B@DC76,3457
630 DATA 0922@E276EBE12B722B732BED5B@E07
    6722B732B22@F47EC9ED5B@DE76,3463
640 DATA ED53@E8762A@E076ED5B@E876CDEABB
    2A@E876ED4B@EA760922@E876545D,3984
650 DATA 018002A7ED42D02A@E076CDF0BB473A
    @ED76B828D5C92A@DE76018002AF,3635
660 DATA ED42301A2A@E076018F01ED4230102A
    @E076ED5B@DE76CDF0BB473A@ED76,3441
670 DATA B8C9FE01C92A@F47E11@F6867DBB200
    37CBAC83A@ED76CDDEBB2A@F47E23,3981
680 DATA 5E2356ED53@E076234E2346E5EB5059
    CDC0BBE1235E2356E52A@E076,3400
690 DATA CDF6BBE111@F6867DBB20D97CBA20D5
    3A@EC76CDDEBB2A@EE76F9C90000,4250
700 DATA END
```


GRASP

The amazing GRASP graph and function plotter. Even better than the 5-star Spectrum version. Watch out for a great review in your official Amstrad User Club mag. Price £8.50 inc p.p.

TOOLBOX. You'll never find better than this for utilities. Contains Graphics and Sprites creator (usable in your own programs), cassette backup utility with high-speed options, and a M/C monitor program. All for just £4.95.

SOFTWARE FOR AMSTRAD CPC464
VALUE FOR MONEY = CAMEL
The equation you'll grow to like!

Cheques/POs to:
CAMEL MICROS, WELL PARK, WILLEYS AVE, EXETER

NUMBER 11

NUMBER 11. Previously econimic simulation games have tended to be dull. Not this one! Great sound and graphics, battle for re-election with a chance for a coalition if you don't earn an overall majority, and grab a knighthood if you can. Outstanding value at £5.50 inc. p.p.

NEWLY RELEASED SOFTWARE

CP-CHESS, 4 LEVEL PLAY, MANY FEATURES. £5.50

HOMEPAK. £7.50
HOME BUDGET, ENERGY CALC ++

THE HOME COMPUTER CENTRE

COMPUTAVISION

4 MARKET STREET ST. AUSTELL TEL 75297



AMSTRAD CPC464 GREEN AND COLOUR VERSIONS ALWAYS AVAILABLE FROM STOCK

DISC DRIVES - PRINTERS - JOYSTICKS MODULATORS - BOOKS

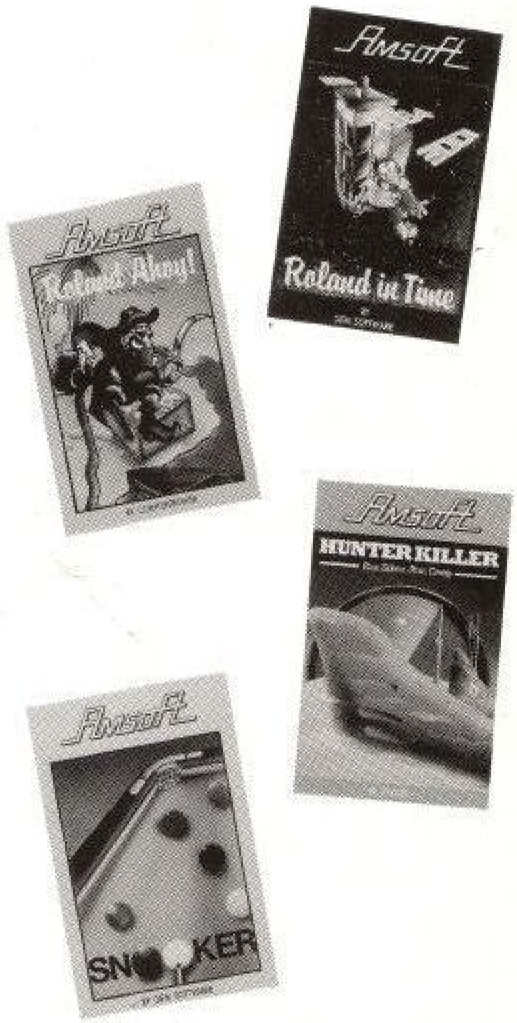
AND THE BEST RANGE

OF SOFTWARE IN THE SOUTH WEST
MAIL ORDER
SEND S.A.E FOR FREE CATALOGUE

SOFTWARE CHART

MONTH ENDING
31-12-84

THIS MONTH	LAST MONTH	SOFTWARE NUMBERS	SOFTWARE TITLES
1	3	SOFT-170	ROLAND IN TIME
2	6	SOFT-192	ROLAND AHOY!
3	N/E	SOFT-190	SNOOKER
4	14	SOFT-135	HUNTER KILLER
5	9	SOFT-154	EASY AMSWORD
6	2	SOFT-129	CODENAME MAT
7	4	SOFT-122	HARRIER ATTACK
8	1	SOFT-117	ROLAND IN THE CAVES
9	5	SOFT-118	ROLAND ON THE ROPES
10	N/E	SOFT-164	ADVANCED AMSWORD



MONITOR EXTENSION MODULE

Computers Today

SEPERATE MONITOR AND KEYBOARD by up to 1.5 metres.

Main AMSTRAD stockists
TOP 50 Amstrad Titles always in stock

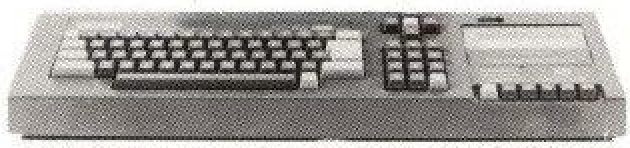




1 PLUGS directly into existing wires
2 Enables monitor to be placed in a more convenient position for word processing
3 MAKES positioning moniter on ADJACENT SHEL possible

MAIL ORDER
31 Market Sq
Bromley, KENT

Tel 01-290-5652



Access & Visa Welcome

£4.99 inc POSTAGE & PACKING [Trade enquire welcome]

UPGRADED TO THE CPC464 AND LEFT YOUR HARDWEAR BEHIND?
RE-CONNECT TO THE OUTSIDE WORLD WITH THE

K.D.S

RS-232 AND PARALLEL INTERFACES

RS-232
Independent RX:TX Baud (75-2400)
Handshake Facilities
Printer Controlled from BASIC
RX Data Buffered to Centronics
Integral Converter for
True RS-232 Voltage Levels
Standard 25 'D' Connector

PARALLEL
Twin 8 bit Ports
Operates direct from BASIC
4 Programmable Operating Modes
Handshake Facilities
An Ideal Unit for Controlling
8 bit printers or the
Robot in Your Life

£45.95 Price incl VAT & P/P **£25.95**

Both units cased and include through connector for interstacking or connection of further add-ons (disc drive etc.)
Literature supplied and software for RS-232 on tape

PROJECTS UNDER DEVELOPMENT
SPEECH SYNTH
SIDEWAYS ROM
MODEM
LIGHT PEN

TEL (04853) 2076

K.D.S. ELECTRONIC'S
15 HILL STREET
HUNSTANTON
NORFOLK
PE36 5BS

PRINTER BARGAINS

VAT & CARRIAGE INCL.

BROTHER M1009 ONLY £178

- 80 col. 50 c.p.s. 9 x 9 dot-matrix
- 196 characters - enlarged, condensed italic, super & subscipt
- Fully Epson compatible
- Friction feed Tractor option at £21
- Roll holder option £8

MANNESMANN TALLY MT80 ONLY £219

- 80 col. 80 c.p.s 9 x 9 dot-matrix
- Quality print style for letter writting
- Tractor and friction feed

OTHER PRINTER BARGAINS

Brother HR5 80 col. 30cps. battery 9 dot matrix	£149
Shinwa CP80 80 col: 80cps	£195
Shinwa CPA80 80 col. 100cps. 2K buffer	£229
Epson RX80FT 80 col. 100cps	£264
Canon PW1080A 80 col. 160cps. Near Letter Quality mode	£319
Kaga Taxan KP810 80 col. 160cps. Near Letter Quality	£299
Daisy Step 2000 daisy 13" platen 18cps	£264

INTERFACES

Parallel Interface cable	£12

SOFTWARE

Tasword 464	£16
Tasprint 464	£9
Tascopy 464	£9

STRONG COMPUTER SYSTEMS.
Bry Cottage, Peniel, Carmarthen, Dyfed. SA32 7DJ.
Tel: 0276 231246 for assistance!!!!!

TIMATIC SYSTEMS LTD

Registered Office
NEWGATE LANE, FAREHAM,
HANTS.PO14 1AN
Tel: FAREHAM (0329) 239953

Sales and Repairs
FAREHAM MARKET
FAREHAM, HANTS
Tel: FAREHAM (0329) 236727

IN STOCK

OVER 90 AMSTRAD CASSETTE TITLES

SOFTWARE FOR CPM SYSTEM
MACRO 80, MICROSOFT BASIC
MICROSOFT BASIC COMPILER
OTHER TITLES ON REQUEST

DISC DRIVES IN STOCK
CPC464 3" £199.95
TIMATIC 5¼" 2ND
DISC DRIVE AVAILABLE

MAIL ORDER WELCOME
S.A.E. FOR FULL LIST

P & P FREE OF CHARGE

ELECTRIC FENCING

This program started out as a version of 'Pong'. However, it got changed half way through its development and now only a whiff of the original idea remains. Electric fencing is a game for two players using either a pair of JY1 joysticks or the keyboard. The object of the game is to score 'hits' on the other player while dodging his attacks. Even though it's written in Basic, the game is quite fast and furious and should provide a few minutes entertaining play. The game contains some useful re-usable routines and interesting sound effects.

The listing is divided into sections to make it easier to understand and illustrate its structure. You are recommended to type in each section and then check it with the listing. Some sections can be left out initially as they not essential to the main programme.
These are:

The title screen. Lines 1490-1610. Replace this whole section with

```
1490 RETURN
```

The joystick/keyboard selection screen. Lines 280-360 Replace this section wi th

```
280 j=0:RETURN'if you only
     wish to use the keyboard
```

If you don't intend to use joysticks then you can dispense with the joystick reading routine. Lines 400-470.

Finally, the 'GAME OVER' routine. Lines 940-1030. Replace this with

```
940 STOP
```

When debugging your program remember that an error (especially `improper argument`) may not be in the line it shows up in. This is something that often confuses people. The computer can 'see' that you're not carrying your umbrella but it can't tell where you left it. If something gets printed in the wrong place, you may have left a semicolon out of the `print` statement. Don't confuse I for 1.

VARIABLES

`p1` and `p2` stand for player one and player two. Used to prefix these variables requiring a separate record for each player;

`dir` for direction `sa` for sword activated `sc` for score `wp` for window position

These other variables are used as switches to alter the flow of control through the program.

`j` for joystick or keyboard control. -1 (or true) for joysticks, 0 for keys. `pwpe` for players window positions are equal.

The string variable `a$(n)` contains the big characters 0-9. Where `n` is the required character.
`pt` for temporary pointer

THE ROUTINES

50-110 Goto the various routines that set up the program

140-240 Main loop that calls the routines to test the keys and update the screen.

280-360 Choose joystick or keyboard control

400-470 Test the joysticks

500-560 Test the keyboard

600-620 and 660-680 Update the players positions on the screen.

720-740 and 780-800 Draw players 'swords'.

840-900 Test for either player being hit and update score board.

940-1030 Game over

1070-1210 Set up string array. Change 1's and 0's into blocks and spaces.

1240-1340 Setup other constants.

1380-1450 Reset variables and screen at start of action.

1490-1610 Title screen.

When you run the program, you should get the title screen appearing and then after a pause you should see the joystick/keyboard option come up. If you want to use joysticks, press j and the asterisk should move to the joystick position. Press **[ENTER]** to make your choice. The title screen should come up once more and then the game starts. If you are using the keyboard the controls are:

PLAYER1	PLAYER2
up a	up 6
down z	down 3
fire x	fire 7

When either player gets a hit on the other player, his score goes up by one and this is displayed in large characters on the screen. To add to the excitement, the sound of a generator can be heard in the background and various other noises should be heard. When a player's score reaches nine, he wins and a key must be pressed to start again.

There are many ways in which you could alter the game. You could perhaps allow the players to move in any direction. Put obstacles on the screen to hide behind. Limit the amount of energy available for moving and shooting or include a high score table.

LISTING

```
10 REM ***********************************
   *****************
20 REM *** Programs from AMSTRAD CPC464
   User Magazine ***
30 REM **********         FENCING       *
   *****************
40 REM ***********************************
   *****************
50 DEFINT a-z
60 MODE 0
70 GOSUB 1070
80 GOSUB 1490
90 GOSUB 280
100 GOSUB 1490
110 GOSUB 1380
120 '
130 '
140 REM start
150 IF finished THEN GOTO 100
160 GOSUB 240
170  CALL &BD19:IF p1dir THEN GOSUB 600
    ELSE CALL &BD19:CALL &BD19
180  CALL &BD19:IF p2dir THEN GOSUB 660
    ELSE CALL &BD19:CALL &BD19
190 IF p1sa=-1 THEN GOSUB 720
200 IF p2sa=-1 THEN GOSUB 780
210 GOTO 140
220 '
230 '
240 IF j THEN  400 ELSE  500
250 '
260 '
270 '
280 CLS:PEN 6
290 PRINT:PRINT"   CHOOSE CONTROL"
300 PRINT:PRINT:PRINT:PRINT" press J K o
    r ENTER"
310 LOCATE 4,10:PRINT"JOYSTICK";TAB(5);"
    OR KEYS"
320 LOCATE 12,10:IF j THEN PRINT"*":ELSE
     PRINT" "
330 LOCATE 12,11:IF j THEN PRINT" ":ELSE
     PRINT"*"
340 IF NOT(INKEY(45)) THEN j=-1
350 IF NOT(INKEY(37)) THEN j=0
360 IF NOT(INKEY(18)) THEN RETURN ELSE 3
    20
370 '
380 '
390 '
400 p1=JOY(0):p2=JOY(1)
410 p1dir=(p1 AND 1)*-1+(p1 AND 2)*0.5
420 p2dir=(p2 AND 1)*-1+(p2 AND 2)*0.5
430 IF P1 AND 16 THEN p1sa=p1sa-1:IF p1s
    a=-1 THEN AFTER 15 GOSUB 840
440 IF P2 AND 16 THEN p2sa=p2sa-1:IF p2s
    a=-1 THEN AFTER 15 GOSUB 840
450 IF p1sa THEN p1dir=0
460 IF p2sa THEN p2dir=0
470 RETURN
480 '
490 '
500 p2dir=((INKEY(4)=0)*1)+((INKEY(5)=0)
    *-1)
510 p1dir=((INKEY(69)=0)*1)+((INKEY(71)=
    0)*-1)
520 IF INKEY(63)=0 THEN p1sa=p1sa-1:IF p
    1sa=-1 THEN AFTER 15 GOSUB 840
530 IF INKEY(10)=0 THEN p2sa=p2sa-1:IF p
    2sa=-1 THEN AFTER 15 GOSUB 840
540 IF p1sa THEN p1dir=0
550 IF p2sa THEN p2dir=0
560 RETURN
570 '
580 '
590 '
600 pt=p1wp+p1dir:IF pt>25 OR pt<6 THEN
    RETURN ELSE p1wp=pt
610 p1dir=0
620 PEN 1:LOCATE 3,p1wp:CLS #3:PRINT CHR
    $(209);:RETURN
630 '
640 '
650 '
660 pt=p2wp+p2dir:IF pt>25 OR pt<6 THEN
    RETURN ELSE p2wp=pt
670 p2dir=0
680 PEN 2:LOCATE 18,p2wp:CLS #5:PRINT CH
    R$(211);:RETURN
690 '
700 '
710 '
720 PAPER #4,4:WINDOW #4,4,17,p1wp,p1wp:
    CLS#4:CALL &BD19:CALL &BD19
730 PAPER #4,0:CLS#4
740 GOTO 600
750 '
760 '
770 '
780 PAPER #6,5:WINDOW #6,4,17,p2wp,p2wp:
    CLS#6:CALL &BD19:CALL &BD19
790 PAPER #6,0:CLS#6
800 GOTO 660
810 '
820 '
830 '
840 pwpe=(p1wp=p2wp):IF p1sa AND NOT(p2s
    a) AND pwpe THEN p1sc=p1sc+1:SOUND 1
    32,120,10,0,1,0:PRINT#1,a$(p1sc);:IF
     p1sc=9 THEN 940
850 IF p2sa AND NOT(p1sa) AND pwpe THEN
    p2sc=p2sc+1:SOUND 132,100,10,0,1,0:P
    RINT#2,a$(p2sc);:IF p2sc=9 THEN 940
860 IF p1sa THEN SOUND 132,40,70,0,1,1
870 IF p2sa THEN SOUND 132,56,70,0,1,1
880 p1sa=0
890 p2sa=0
900 RETURN
```

```
910 '
920 '
930 '
940 PEN 6
950 LOCATE 6,10:PRINT"GAME OVER"
960 IF p1sc=9 THEN INK 1,2,20:INK 2,0 EL
    SE INK 2,6,17:INK 1,0
970 SOUND 129,1000,0,12,3:SOUND 130,900,
    0,12,3
980 WHILE INKEY$<>"":WEND
990 t!=TIME:WHILE t!+2000>TIME:WEND
1000 WHILE INKEY$="":WEND
1010 CLS
1020 finished=-1
1030 RETURN
1040 '
1050 '
1060 '
1070 a$(0)="111101101101111"
1080 a$(1)="011001001001001"
1090 a$(2)="111001111100111"
1100 a$(3)="111001111001111"
1110 a$(4)="100100101111001"
1120 a$(5)="111100111001111"
1130 a$(6)="111100111101111"
1140 a$(7)="111001001010010"
1150 a$(8)="111101111101111"
1160 a$(9)="111101111001001"
1170 FOR n=0 TO 9
1180  howlong=LEN(a$(n))
1190  FOR n2=1 TO howlong
1200    IF MID$(a$(n),n2,1)="1"THEN MID$
     (a$(n),n2,1)=CHR$(143)ELSE MID$(a$(
     n),n2,1)=CHR$(32)
1210 NEXT n2,n
1220 '
1230 '
1240 b$="ELECTRIC FENCING"
1250 c$=CHR$(32)+CHR$(164)+" Alexander M
     artin"
1260 ENV 1,=9,2000:ENT -1,6,3,1
1270 ENV 2,127,0,0,127,0,0,127,0,0,127,0
     ,0,127,0,0
1280 ENV 3,=9,9000
1290 '
1300 '
1310 '
1320 BORDER 0
1330 PEN #4,1:PEN #6,2:PEN #1,1:PEN #2,2
     :PAPER #1,3:PAPER #2,3:PEN #0,6
1340 RETURN 'FROM SETTING UP CONSTANTS
1350 '
1360 '
1370 '
1380 INK 0,12:INK 1,2:INK 2,6:INK 3,13:I
     NK 4,20:INK 5,17:INK 6,20
1390 WINDOW #3,3,3,6,25:WINDOW #5,18,18,
     6,25
1400 WINDOW #1,3,5,1,5:WINDOW #2,16,18,1
     ,5:WINDOW #7,1,20,1,5:PAPER #7,3
1410 CLS:CLS#7:PRINT#1,a$(0);:PRINT#2,a$
     (0);:p1sc=0:p2sc=0:p1wp=5:p2wp=24:p
     1dir=1:p2dir=1
1420 GOSUB 600:GOSUB 660
1430 SOUND 1,1000,0,12,2:SOUND 2,900,0,1
     2,2
1440 p1sa=0:p2sa=0:finished=0
1450 RETURN 'FROM GAME SHEET RESTORE
1460 '
1470 '
1480 '
1490 CLS
1500 PEN 7
1510 FOR n=1 TO LEN(b$)
1520  LOCATE 2+n,10
1530  FOR n2=LEN(b$) TO n STEP-1
1540   PRINT MID$(b$,n2,1)
1550   LOCATE 2+n,10
1560   SOUND 135,20*n2,5,12,2,1
1570 NEXT n2,n
1580 SOUND 135,100,0,13,3,1,20
1590 PEN 6:PRINT:PRINT:PRINT:PRINT c$
1600 t!=TIME:WHILE t!+2500>TIME:WEND
1610 RETURN
```



CENTRE COURT
You CANNOT be serious!
A most accomplished tennis game for the CPC 464 for only £8.95
SELECT
F TO PLAY WITH A FRIEND
M TO PLAY THE COMPUTER <ARNOLD>
R FOR DEMO ARNOLD V. ROLAND
AMSTRAD CPC464
Serve yourself another ace from Amsoft!

Flash Man

Brief Summary

This program is in two sections. The first comprises lines 10-840 and draws the computer and colour monitor as an introduction to the second part of the program which is the game itself.

The program could be shortened if desired by deleting lines 10-870. Lines 880-2660 comprises the game of "FLASHMAN". The player moves a figure about the screen using the cursor keys or a joystick.

The object is to reach the top of the screen, starting from the bottom before 'Flashman' is hit by randomly produced 'cosmic rays'. These rays also clear the way through a block structure for Flashman to move (he cannot move into a square coloured black).

Points are awarded throughout the game which has five levels of various sounds and three melodies. Use is made of interrupt commands (EVERY/DI/EI) as well as WHILE/WEND.

Full instructions are printed at the start of the game.

Program in Detail

Line 50	Left in for convenience, programs decimal point key for listing purposes.
Lines 60-850	Draw computer/monitor and introductory heading.
Line 860	Produces different coloured inks on the monitor screen in the drawing.
Lines 890-1060	Draws introductory screen plays a tune. Draws instruction screens.
Lines 960-1000	Establishes starting point for various levels of game.
Lines 1010-1230	Draws in main screen (black block structure on white back ground, flashing character, score, fazers left). Establishes interrupt routine for cosmic rays. Plays another tune prior to start of play.
Lines 1250-1700	Main loop. Controls character movement, using EI/DI, 'TEST' for limiting movement, tests for reaching top of screen, etc. Score altered with movements.
Lines 1720-1800	'Cosmic Rays' routine. Use of 'Z' variable in first line allows avoidence of routine when denied at certain points in the program. Random production of 20 locations on screen. Red ink and sound to represent ray bombardement. Test to see if ray has hit 'Flashman', or to jumps the routine below.
Lines 1820-1840	'Failure' routine. Interrupt avoided (Z=0), while / wend loop completed (E=21). Explosion sound and red/yellow inks. "Sinking" sound. Program re-started at line 960.
Lines 1860-1980	'Success' routine. Interrupt has been avoided (Z = 0) in line 1510. 'Winning' sound. Screen prints up 'YOU MADE IT'. Bonus score 50 points added. Data for second melody 'restored' before game restarted at next level or if already at level 5 -goes to "Home" routine (below).
Lines 2000-2020	Routine called from early in program, draws opening screen, etc.
Lines 2040-2180	Routine called from early in program, draws instruction screens, etc.
Lines 2200-2420	'Home' routine. Draws find screen. Adds bonus 500 points to score, prints find score. Gives option to play again if desired or end program.
Lines 2440-2660	Three sound subroutines as called from earlier in the program plus DATA. Some envelopes were defined at the beginning of the program.

```
10 '**************************************
   *************************************
20 '********** Programs from issue 4 of
   CPC464 User Magazine ***************
30 '*********************** 'Amstrad CPC
    464 drawing' **********************
40 '**************************************
   *************************************
50 KEY 138,"MODE 2:PEN 1:INK 1,26:ink 0,
   13:border 13:list"+CHR$(13)
60 MODE 0:BORDER 14:INK 0,14:INK 6,11:IN
   K 2,20:INK 3,6
70 PEN 2:LOCATE 2,2:PRINT"AMSTRAD":PEN 1
   3:LOCATE 2,5:PRINT"CPC 464":PEN 11:LO
   CATE 2,8:PRINT"presents..."
80 FOR y=1 TO 146:PLOT 68,y:DRAW 572,y,8
   :NEXT
90 y=140
100 FOR x=92 TO 548 STEP 12:PLOT x,y,5:N
    EXT
110 y=y+1:IF y>146 THEN 120 ELSE 100
120 PLOT 102,124:DRAW 342,124:DRAW 342,1
    04:DRAW 102,104:DRAW 102,122
130 PLOT 112,108,4:DRAW 112,121:DRAW 120
    ,121:DRAW 120,108:PLOT 116,112 :PLOT
     128,108:DRAW 128,121:DRAW 144,121:D
    RAW 144,108:PLOT 136,118:DRAW 136,10
    8 :PLOT 152,108:DRAW 160,108:DRAW 16
    0,115:DRAW 152,115:DRAW 152,121:DRAW
     160,121
140 PLOT 168,121:DRAW 176,121:PLOT 172,1
    19:DRAW 172,108 :PLOT 184,108:DRAW 1
    84,121:DRAW 192,121:DRAW 192,115:DRA
    W 188,115:DRAW 192,108 :PLOT 200,108
    :DRAW 200,121:DRAW 208,121:DRAW 208,
    108:PLOT 204,112
150 PLOT 216,108:DRAW 216,121:DRAW 220,1
    21:DRAW 224,119:DRAW 224,110:DRAW 22
    0,108
160 PLOT 102,92,5:DRAW 342,92:DRAW 342,2
    8:DRAW 308,28:DRAW 308,12:DRAW 150,1
    2:DRAW 150,28:DRAW 102,28:DRAW 102,9
    0 : PLOT 102,90:DRAW 342,90:PLOT 102
    ,26:DRAW 342,26

170 PLOT 150,28:DRAW 304,28:PLOT 150,14:
    DRAW 304,14:PLOT 102,77:DRAW 342,77:
    PLOT 102,75:DRAW 342,75:PLOT 102,59:
    DRAW 316,59:PLOT 102,61:DRAW 316,61:
    PLOT 102,43:DRAW 342,43:PLOT 102,45:
    DRAW 342,45:PLOT 294,15:DRAW 294,25
180 y=79
190 FOR x=116 TO 324 STEP 16:PLOT x,y:NE
    XT
200 y=y+1:IF y>89 THEN 210 ELSE 190
210 y=63
220 FOR x=120 TO 312 STEP 16:PLOT x,y:NE
    XT
230 y=y+1:IF y>73 THEN 240 ELSE 220
240 y=47
250 FOR x=124 TO 316 STEP 16:PLOT x,y:NE
    XT
260 y=y+1:IF y>57 THEN 270 ELSE 250
270 y=31
280 FOR x=132 TO 308 STEP 16:PLOT x,y:NE
    XT
290 y=y+1:IF y>41 THEN 300 ELSE 280
300 PLOT 330,120:DRAW 330,117,3:PLOT 104
    ,78:DRAW 104,89:PLOT 108,78:DRAW 108
    ,89:PLOT 112,78:DRAW 112,89
310 y=63
320 FOR x=104 TO 116 STEP 4:PLOT x,y,12:
    NEXT
330 y=y+1:IF y>73 THEN 340 ELSE 320
340 y=47
350 FOR x=104 TO 120 STEP 4:PLOT x,y:NEX
    T
360 y=y+1:IF y>57 THEN 370 ELSE 350
370 y=31
380 FOR x=104 TO 128 STEP 4:PLOT x,y:NEX
    T
390 y=y+1:IF y>41 THEN 400 ELSE 380
400 y=17
410 FOR x=296 TO 304 STEP 4:PLOT x,y:NEX
    T
420 y=y+1:IF y>25 THEN 430 ELSE 410
430 y=31
440 FOR x=312 TO 336 STEP 4:PLOT x,y:NEX
    T
450 y=y+1:IF y>41 THEN 460 ELSE 440
460 y=78
470 FOR x=328 TO 336 STEP 4:PLOT x,y:NEX
    T
480 y=y+1:IF y>89 THEN 490 ELSE 470
490 y=47
500 FOR x=320 TO 336 STEP 4:PLOT x,y,6:N
    EXT
510 y=y+1:IF y>73 THEN 520 ELSE 500
520 PLOT 316,63:DRAW 316,73
530 y=10
540 FOR x=368 TO 416 STEP 16:PLOT x,y,5:
    NEXT
550 y=y+1:IF y>76 THEN 560 ELSE 540
560 PLOT 372,10:DRAW 412,10:PLOT 372,12:
    DRAW 412,12:PLOT 372,26:DRAW 412,26:
    PLOT 372,28:DRAW 412,28:PLOT 372,42:
    DRAW 412,42:PLOT 372,44:DRAW 412,44:
    PLOT 372,58:DRAW 412,58:PLOT 372,60:
    DRAW 412,60:PLOT 372,74:DRAW 412,74
570 PLOT 372,76:DRAW 412,76
580 y=14
590 FOR x=404 TO 412 STEP 4:PLOT x,y,6:N
    EXT
600 y=y+1:IF y>25 THEN 610 ELSE 590
610 PLOT 384,84:DRAW 384,132,5:DRAW 400,
    132:DRAW 400,84:DRAW 388,84:DRAW 388
    ,86:DRAW 396,86:PLOT 384,134:DRAW 40
    0,134:PLOT 368,100:DRAW 368,118:DRAW
     416,118:DRAW 416,100:DRAW 372,100:P
    LOT 372,102:DRAW 414,102:PLOT 372,11
    6:DRAW 412,116
620 y=104
630 FOR x=388 TO 396 STEP 4:PLOT x,y,6:N
    EXT
640 y=y+1:IF y>114 THEN 650 ELSE 630
650 PLOT 448,4:DRAW 544,4,5:PLOT 448,6:D
    RAW 544,6:DRAW 544,23:DRAW 448,23:DR
    AW 448,7:PLOT 452,21:DRAW 540,21
660 y=7
670 FOR x=464 TO 528 STEP 16:PLOT x,y:NE
    XT
680 y=y+1:IF y>20 THEN 690 ELSE 670
690 PLOT 440,1:DRAW 440,26:DRAW 552,26:D
    RAW 552,1
```

```
700 y=9
710 FOR x=452 TO 460 STEP 4:PLOT x,y,3:N
    EXT
720 y=y+1:IF y>19 THEN 730 ELSE 710
730 PLOT 440,30:DRAW 552,30,5:DRAW 552,5
    0:DRAW 440,50:DRAW 440,32
740 PLOT 440,54:DRAW 440,118:DRAW 552,11
    8:DRAW 552,54:DRAW 442,54:PLOT 452,7
    6:DRAW 452,110:DRAW 540,110:DRAW 540
    ,76:DRAW 454,76
750 PLOT 452,122:DRAW 472,122:DRAW 472,1
    30:DRAW 452,130:DRAW 452,124
760 FOR y=175 TO 182:PLOT 392,y:DRAW 622
    ,y,8:NEXT
770 FOR y=188 TO 395:PLOT 380,y:DRAW 632
    ,y,8:NEXT
780 PLOT 378,399,5:DRAW 378,184:DRAW 636
    ,184:DRAW 636,399:DRAW 382,399:DRAW
    382,397:DRAW 632,397:PLOT 382,186:DR
    AW 632,186
790 PLOT 378,399,5:DRAW 378,184:DRAW 636
    ,184:DRAW 636,399:DRAW 382,399:DRAW
    382,397:DRAW 632,397:PLOT 382,186:DR
    AW 632,186:PLOT 382,202:DRAW 632,202
    : PLOT 624,194:PLOT 624,192
800 PLOT 392,218:DRAW 620,218:DRAW 620,3
    84:DRAW 392,384:DRAW 392,220:PLOT 39
    4,382:DRAW 618,382:PLOT 392,216:DRAW
     620,216
810 PLOT 388,182:DRAW 388,172:DRAW 624,1
    72:DRAW 624,182:PLOT 400,170:PLOT 40
    4,170:PLOT 400,168:PLOT 404,168:PLOT
     400,166:PLOT 404,166:PLOT 608,170:P
    LOT 612,170:PLOT 608,168:PLOT 612,16
    8:PLOT 608,166:PLOT 612,166
820 FOR y=231 TO 370:PLOT 408,y:DRAW 604
    ,y:NEXT
830 PLOT 480,178:DRAW 480,166:DRAW 472,1
    74:DRAW 472,166:DRAW 464,174:DRAW 46
    4,166:DRAW 456,174:DRAW 456,166:DRAW
     448,174:DRAW 448,166:DRAW 440,174:D
    RAW 440,166:DRAW 432,174:DRAW 432,16
    6:DRAW 424,174:DRAW 424,166:DRAW 416
    ,174:DRAW 416,148
840 PLOT 492,178:DRAW 492,154:DRAW 484,1
    62:DRAW 484,154:DRAW 476,162:DRAW 47
    6,154:DRAW 468,162:DRAW 468,154:DRAW
     460,162:DRAW 460,154:DRAW 452,162:D
    RAW 452,154:DRAW 444,162:DRAW 444,15
    4:DRAW 436,162:DRAW 436,154:DRAW 428
    ,162:DRAW 428,148
850 PLOT 488,160:DRAW 432,160
860 FOR p=0 TO 12:FOR y=231 TO 370:PLOT
    408,y,p:DRAW 604,y:NEXT y:NEXT p
870 FOR t=1 TO 1000:NEXT
880 '************************ 'FLASHMAN'
     by P.J.Eva ************************
890 ENT -1,2,1,4,2,-1,4:ENT -2,50,40,2:E
    NT -3,1,-30,10,1,30,10
900 SPEED INK 10,10
910 GOSUB 2000
920 GOSUB 2440
930 FOR t=1 TO 1500:NEXT
940 GOSUB 2040
950 '********************************* se
    t up ********************************
    **
960 L=1:s=0:f=3:GOTO 1010
970 L=2:GOTO 1010
980 L=3:GOTO 1010
990 L=4:GOTO 1010
1000 L=5:GOTO 1010
1010 MODE 1:BORDER 13:INK 0,26:INK 1,6:I
     NK 2,18,8:INK 3,0
1020 GOSUB 2470
1030 FOR q=1 TO 23 STEP 2
1040 PEN 3:LOCATE 1,q:PRINT STRING$(40,C
     HR$(143)):NEXT
1050 IF L=5 THEN 1090 ELSE 1060
1060 IF L=4 THEN 1100 ELSE 1070
1070 IF L=3 THEN 1110 ELSE 1080
1080 IF L=2 THEN 1120 ELSE 1140
1090 LOCATE 1,22:PRINT STRING$(40,CHR$(1
     43)):LOCATE 1,20:PRINT STRING$(40,C
     HR$(143))
1100 LOCATE 1,18:PRINT STRING$(40,CHR$(1
     43)):LOCATE 1,16:PRINT STRING$(40,C
     HR$(143)):LOCATE 1,14:PRINT STRING$
     (40,CHR$(143))
1110 LOCATE 1,12:PRINT STRING$(40,CHR$(1
     43)):LOCATE 1,10:PRINT STRING$(40,C
     HR$(143)):LOCATE 1,8:PRINT STRING$(
     40,CHR$(143))
1120 LOCATE 1,6:PRINT STRING$(40,CHR$(14
     3)):LOCATE 1,4:PRINT STRING$(40,CHR
     $(143)):LOCATE 1,2:PRINT STRING$(40
     ,CHR$(143))
1130 IF L=5 GOTO 1160
1140 FOR d=1 TO 40 STEP 3:FOR g=2 TO 22
     STEP 2
1150 LOCATE d,g:PRINT CHR$(143):NEXT g:N
     EXT d
1160 EVERY 100 GOSUB 1720
1170 x=20:y=24
1180 LOCATE 3,25:PRINT"score":LOCATE 29,
     25:PRINT"fazers"
1190 GOSUB 1680
1200 SOUND 1,40,10,6
1210 GOSUB 1690
1220 GOSUB 1700
1230 z=1
1240 '********************** character m
     ovement routine *******************
     ***
1250 FOR t=1 TO 50:NEXT
1260 IF INKEY(9)=0 AND y-1>0 OR JOY(0)=1
     6 AND y-1>0 THEN 1310 ELSE 1270
1270 IF INKEY(0)=0 OR JOY(0)=1 THEN 1350
      ELSE 1280
1280 IF INKEY(2)=0 OR JOY(0)=2 THEN 1450
      ELSE 1290
1290 IF INKEY(8)=0 OR JOY(0)=4 THEN 1540
      ELSE 1300
1300 IF INKEY(1)=0 OR JOY(0)=8 THEN 1610
      ELSE 1260
1310 IF f<1 THEN 1270
1320 DI:PEN 0:LOCATE x,y-1:PRINT CHR$(14
     3):SOUND 1,10,8,7
1330 f=f-1:GOSUB 1700
1340 EI:FOR t=1 TO 200:NEXT:GOTO 1270
1350 IF TEST((x*16-8),((25-y)*16+24))=3
     THEN 1260
1360 y=y-1
```

```
1370 s=s+10:GOSUB 1690
1380 IF y<1 THEN 1390 ELSE 1400
1390 z=0:GOTO 1860
1400 DI
1410 PEN 1:LOCATE x,y+1:PRINT CHR$(32)
1420 GOSUB 1680
1430 EI
1440 GOTO 1250
1450 IF TEST((x*16-8),((25-y)*16-8))=3 T
     HEN 1260
1460 IF y<24 THEN 1470 ELSE 1260
1470 y=y+1:s=s-15
1480 GOSUB 1690
1490 DI
1500 PEN 1:LOCATE x,y-1:PRINT CHR$(32)
1510 GOSUB 1680
1520 EI
1530 GOTO 1250
1540 IF TEST((x*16-24),((25-y)*16+8))=3
     THEN 1260
1550 x=x-1:IF x<1 THEN x=1
1560 DI
1570 PEN 1:LOCATE x+1,y:PRINT CHR$(32)
1580 GOSUB 1680
1590 EI
1600 GOTO 1250
1610 IF TEST((x*16+8),((25-y)*16+8))=3 T
     HEN 1260
1620 x=x+1:IF x>40 THEN x=40
1630 DI
1640 PEN 1:LOCATE x-1,y:PRINT CHR$(32)
1650 GOSUB 1680
1660 EI
1670 GOTO 1250
1680 PEN 2:LOCATE x,y:PRINT CHR$(225):RE
     TURN
1690 PEN 1:LOCATE 9,25:PRINT s:RETURN
1700 PEN 1:LOCATE 36,25:PRINT f:RETURN
1710 '**************************** cosmic
      rays ****************************
1720 IF z=1 THEN 1740 ELSE 1730
1730 RETURN
1740 e=1:WHILE e<21
1750 a=INT(RND*40)+1:b=INT(RND*24)+1
1760 PEN 1:LOCATE a,b:PRINT CHR$(143):e=
     e+1
1770 SOUND 1,0,3,6,0,0,30
1780 LOCATE a,b:PRINT CHR$(32)
1790 IF a=x AND b=y THEN 1820
1800 WEND:RETURN
1810 '************************* failure r
     outine ************************
1820 z=0:e=21:SOUND 1,0,100,7,0,0,5:INK
     0,6,24:BORDER 24,6:FOR t=1 TO 1000:
     NEXT
1830 SOUND 1,300,190,7,0,2:FOR t=1 TO 14
     00:NEXT
1840 RUN 960
1850 '************************ success ro
     utine ***************************
1860 MODE 0:INK 0,3:INK 1,24:LOCATE 5,12
     :PRINT"YOU MADE IT !"
1870 SOUND 1,119,150,7,0,3
1880 s=s+50:IF L<5 THEN RESTORE 2580
1890 FOR t=1 TO 1500:NEXT
1900 IF L=5 THEN 2200
1910 IF L=4 THEN 1950
1920 IF L=3 THEN 1960
1930 IF L=2 THEN 1970
1940 IF L=1 THEN 1980
1950 GOTO 1000
1960 GOTO 990
1970 GOTO 980
1980 GOTO 970
1990 '***************************** openi
     ng screen *************************
     **
2000 MODE 0:INK 0,0:BORDER 0:PLOT 32,32,
     6:DRAW 608,32:DRAW 608,368:DRAW 32,
     368:DRAW 32,32
2010 INK 1,18,6:INK 3,8:INK 2,2:PEN 1:LO
     CATE 7,12:PRINT"FLASHMAN":PEN 2:LOC
     ATE 7,13:PRINT"--------":PEN 3:LOCA
     TE 3,22:PRINT"[";CHR$(164);" P.J.Ev
     a 1984]"
2020 RETURN
2030 '*************************** instruc
     tions ***************************
2040 MODE 1:BORDER 1:INK 0,1:INK 1,24:IN
     K 2,15:INK 3,17
2050 PEN 2:LOCATE 9,1:PRINT"I N S T R U
     C T I O N S":LOCATE 9,2:PRINT STRIN
     G$(23,"-")
2060 PEN 1:PRINT:PRINT"  FLASHMAN is stu
     ck at the bottom of thescreen.He de
     sperately wants to get up tothe top
      of the screen but in his way arebl
     ock structures.":PRINT" Fortunately
      FLASHMAN has the power to"
2070 PRINT"cause cosmic rays to bombarde
      the block structures and create ga
     ps for him to   pass through.These
     rays,however,strike  completely at
     random and if they hit    FLASHMAN
     he is dead."
2080 PRINT" FLASHMAN also carries 3 Faze
     rs, each iscapable of blasting away
      one block in   front of him.Fazers
      can be used at any  time (by press
     ing the COPY key or joy-  stick FIR
     E button) but should be kept inrese
     rve for tougher levels,if possible.
     "
2090 PEN 3:PRINT:PRINT"(press any key to
      continue instructions)"
2100 IF INKEY$="" THEN 2100
2110 CLS:PEN 3:PRINT" Your task is to gu
     ide FLASHMAN,using   the cursor key
     s or joystick, through thegaps blas
     ted between the blocks to       safe
     ty at the top of the screen."
2120 PEN 1:PRINT:PRINT" Points are award
     ed as follows:-":PRINT:PRINT"    Up
     ward movement    ";:PEN 2:PRINT"+10"
     ;:PEN 1:PRINT" points.":PRINT"    D
     ownward movement";:PEN 2:PRINT" -15
     ";:PEN 1:PRINT" points."
2130 PRINT"    Reaching safety    ";:PEN
     2:PRINT"+50";:PEN 1:PRINT" points."
2140 PEN 3:PRINT:PRINT"Sadly,safety,once
      reached is short livedand FLASHMAN
      soon realises that the top of one
     screen is merely the bottom of   an
     other  more difficult screen!"
```

SHINECREST

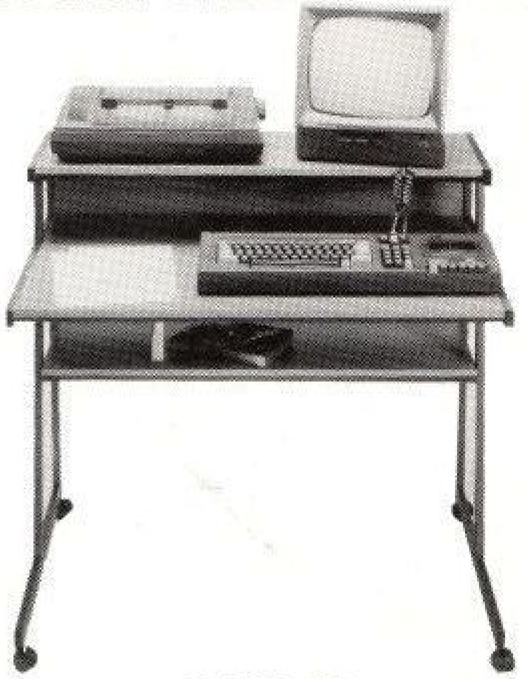
COMPUTER WORKSTATIONS AND DESKS

The latest addition to the Shinecrest range has been enthusiastically received by both dealers and home computer owners for providing all the facilities a computer user could require. Shown in our photograph with the Amstrad CPC464 you can see there remains plenty of space both behind and alongside the computer. The raised shelf provides ample room for your monitor, and if you choose, a printer. Any weight will be easily supported as all the shelves are fitted to two metal rails which travel the full width of the desk and locate into the metal ends. Castors provide an often needed mobility for your desk. The shelves are supplied in a Teak woodgrain finish to compliment the dark metalic brown painted metal rails and end frames.

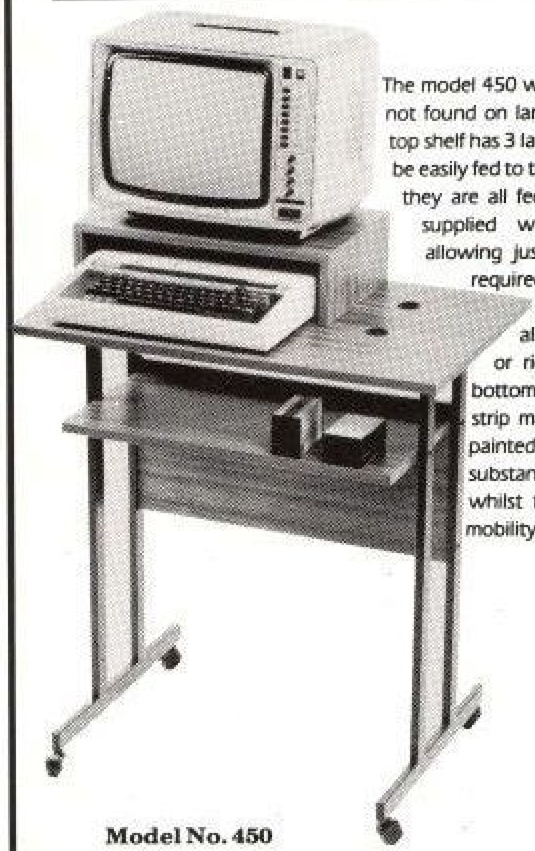
At the back of the desk there is ample space above and below the courtesy panel for interconnecting leads and paper feed to the printer. Every requirement has been provided for.

Dimensions		
	Distance between top and middle shelf	146 mm (5 ³/₄")
	Distance between middle and bottom shelf	216 mm (8 ¹/₂")
	Depth of plinth	292 mm (11 ¹/₂")
	Depth of bottom shelf	320 mm (12 ¹/₂")
	Width	983 mm (36 ³/₄")
	Depth	610 mm (24")
	Height	889 mm (35")

PRICE £69-95

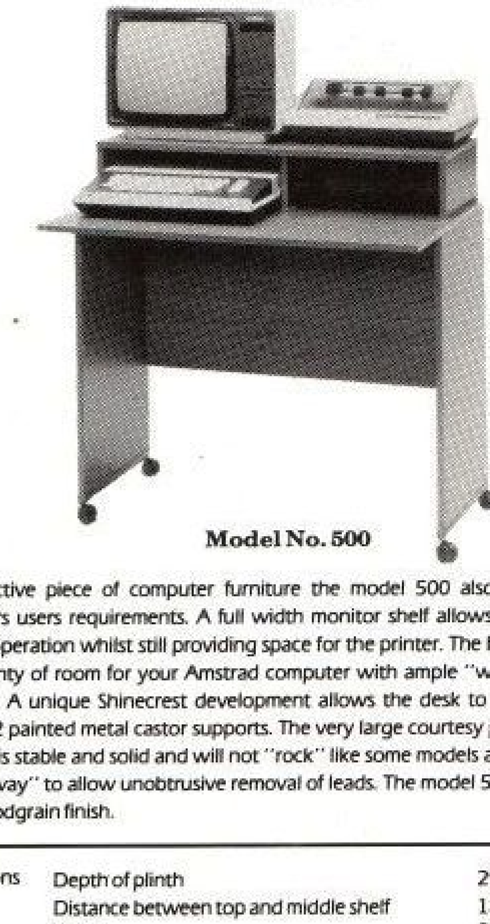


Model No. 600



The model 450 workstation has many features not found on large desks twice the cost. The top shelf has 3 large holes to allow the leads to be easily fed to the back of the unit. Once here they are all fed into a 4 into 1 masterplug supplied with the workstation, thus allowing just one lead (supplied) to be required for mains electricity. The pre formed plinth is moveable allowing it to be used either left or right "handed". Fitted to the bottom edges is a non slip rubber strip making your monitor safe. The painted metal frame provides substantial strength and stability whilst the fitted castors allow easy mobility.

Model No. 450



Model No. 500

An attractive piece of computer furniture the model 500 also provides all the computers users requirements. A full width monitor shelf allows easy left or right handed operation whilst still providing space for the printer. The large work surface gives plenty of room for your Amstrad computer with ample "working room" still available. A unique Shinecrest development allows the desk to be mobile by the fixing of 2 painted metal castor supports. The very large courtesy panel ensures that the desk is stable and solid and will not "rock" like some models available. The back is "cut away" to allow unobtrusive removal of leads. The model 500 is available in a Teak woodgrain finish.

Dimensions		
	Depth of plinth	296 mm (11 ¹/₂")
	Width of plinth	478 mm (18 ³/₄")
	Distance between lower shelf and working shelf	178 mm (7")
	Width	762 mm (30")
	Depth	500 mm (19 ³/₄")
	Height	743 mm (29 ¹/₄")

PRICE £59-95

Dimensions		
	Depth of plinth	296 mm (11 ¹/₂")
	Distance between top and middle shelf	130 mm (5")
	Width	965 mm (38")
	Depth	590 mm (23 ¹/₄")
	Height	865 mm (34")

PRICE £54-95

Shinecrest Products Ltd. are a well established company supplying many major Electronic companys including Amstrad. Our range of products are sold by such outlets as Harrods, Selfridges and D.H. Evans.

To obtain your computer desk or work station we suggest you purchase at any of the following dealers:-

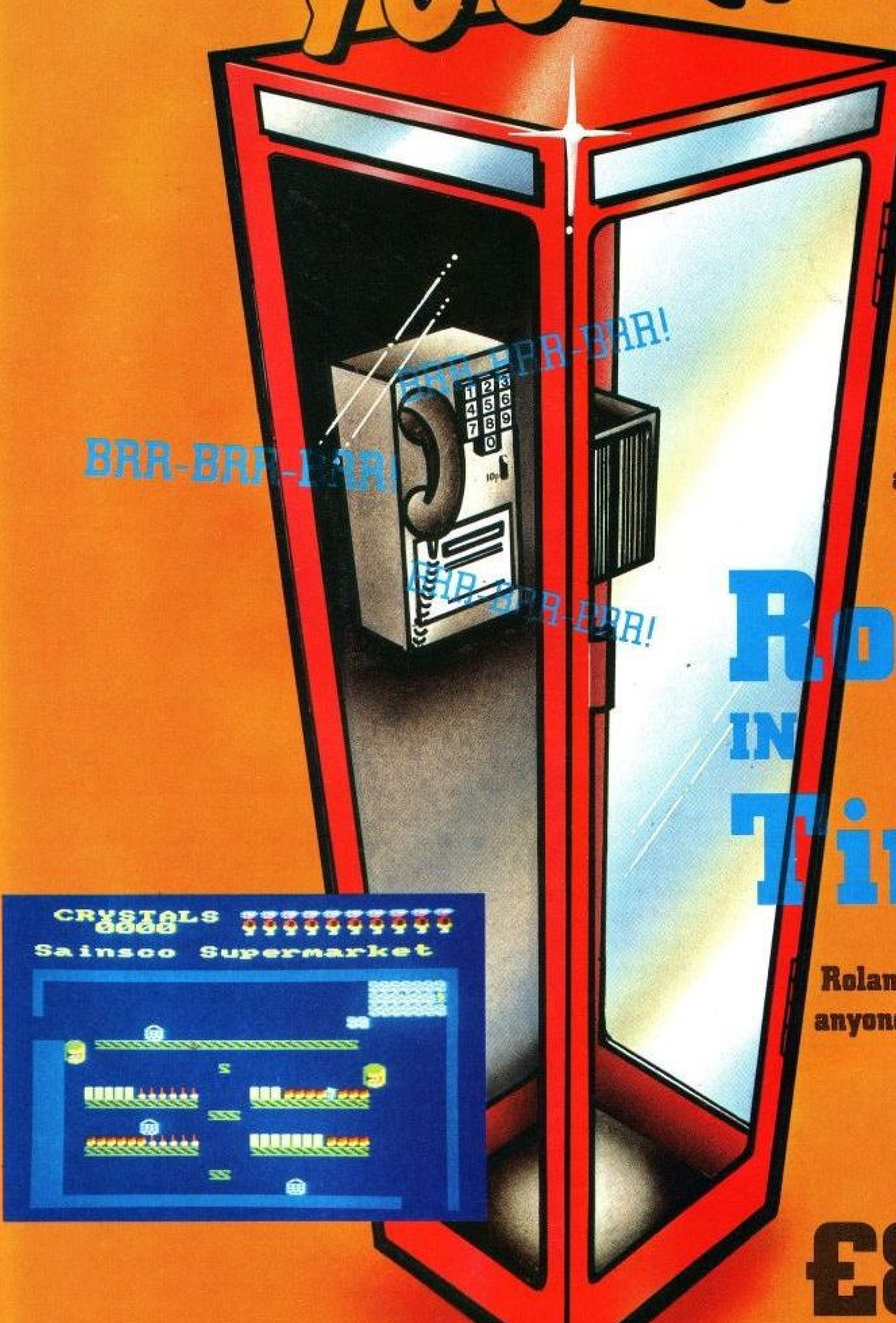
Spectrum Dealers, 3D Computers, Ramco, Sewards, P.H. Electronics, Colchester Co-ops, Oxford & Swindon Co-ops, Trionic, and from most AMSTRAD Stockists.

In the unlikely event that you are unable to obtain the product you require, please contact Shinecrest at the address below, who will provide the name and address of your nearest stockist.

SHINECREST PRODUCTS LTD.,
UNIT 14, DICKER MILL,
HERTFORD, HERTS
TEL. HERTFORD (0992) 51977

```
2150 PEN 1:PRINT:PRINT" You must continu
     e up through each of   the 5 levels
      until you finally reach    home (a
     nd ";:PEN 2:PRINT"500";:PEN 1:PRINT
     " bonus points)."
2160 PEN 2:LOCATE 12,22:PRINT"G O O D
     L U C K !":PEN 3:LOCATE 9,24:PRINT"
     (press any key to start)"
2170 IF INKEY$="" THEN 2170
2180 RETURN
2190 '***************************** home s
     creen *****************************
2200 s=s+500
2210 MODE 1:BORDER 13:INK 0,1:INK 1,1:IN
     K 2,17:INK 3,6
2220 PLOT 16,256,3:DRAW 624,256
2230 n$=CHR$(143)
2240 PEN 1:LOCATE 6,2:PRINT n$;n$;"  ";n
     $;n$;"   ";n$;n$;n$;"   ";n$;n$;"
      ";n$;n$;" ";n$;n$;n$;n$;n$;n$;n$
2250 LOCATE 6,3:PRINT n$;n$;"  ";n$;n$;"
       ";n$;n$;" ";n$;n$;"  ";n$;n$;n$;"
      ";n$;n$;n$;"  ";n$;n$;"   ";n$
2260 LOCATE 6,4:PRINT n$;n$;"  ";n$;n$;"
      ";n$;n$;"   ";n$;n$;" ";n$;n$;n$;n
     $;n$;n$;n$;"  ";n$;n$;" ";n$
2270 LOCATE 6,5:PRINT n$;n$;n$;n$;n$;n$;
     " ";n$;n$;"   ";n$;n$;" ";n$;n$;n$;
     n$;n$;n$;n$;"  ";n$;n$;n$;n$
2280 LOCATE 6,6:PRINT n$;n$;"  ";n$;n$;"
      ";n$;n$;"   ";n$;n$;" ";n$;n$;" ";
     n$;" ";n$;n$;"  ";n$;n$;" ";n$
2290 LOCATE 6,7:PRINT n$;n$;"  ";n$;n$;"
       ";n$;n$;" ";n$;n$;"  ";n$;n$;"
     ";n$;n$;"  ";n$;n$;"   ";n$
2300 LOCATE 6,8:PRINT n$;n$;"  ";n$;n$;"
        ";n$;n$;n$;"   ";n$;n$;"   ";n$;
     n$;" ";n$;n$;n$;n$;n$;n$;n$
2310 INK 1,24
2320 PEN 2:PRINT:PRINT:PRINT:PRINT"  Wel
     l done!   You have successfully
      guided FLASHMAN back to his sanctu
     ary."
2330 PRINT:PRINT:PRINT"  Now he can rela
     x for a while before    venturing o
     ut again into his hostile    enviro
     nment."
2340 PRINT:PRINT:PRINT"  Your final scor
     e is";:PEN 3:PRINT s;:PEN 2:PRINT"p
     oints."
2350 PEN 1:PRINT:PRINT::PRINT"         [
     Another game? (y/n) ]"
2360 PLOT 1,1,3:DRAW 638,1:DRAW 638,398:
     DRAW 1,398:DRAW 1,1:PLOT 3,1:DRAW 3
     ,398:PLOT 636,1:DRAW 636,398
2370 GOSUB 2510
2380 k$=INKEY$
2390 IF k$="y" THEN 2410 ELSE 2400
2400 IF k$="n" THEN 2420 ELSE 2380
2410 RUN 910
2420 PEN 4:INK 4,1:LOCATE 2,23:END
2430 '****************** tunes (sound)
     routines ********************
2440 READ t,d,t2,d2,t4,d4:IF t=999 OR t2
     =999 OR t4=999 THEN RETURN
2450 SOUND 1,t,d,7,0,1:SOUND 2,t2,d2,6,0
     ,1:SOUND 4,t4,d4,5,0,1
2460 GOTO 2440
2470 READ to1,du1,to2,du2,to4,du4
2480 IF to1=999 OR to2=999 OR to4=999 TH
     EN RETURN
2490 SOUND 1,to1,du1,7:SOUND 2,to2,du2,6
     :SOUND 4,to4,du4,4
2500 GOTO 2470
2510 READ t,d,t2,d2,t4,d4:IF t=999 OR t2
     =999 OR t4=999 THEN RETURN
2520 SOUND 1,t,d,7:SOUND 2,t2,d2,7:SOUND
      4,t4,d4,7
2530 GOTO 2510
2540 DATA 1517,50,379,50,47,50,1276,20,3
     19,20,40,20,1517,50,379,50,47,50,0,
     10,0,10,0,10,1517,50,379,50,47,50,1
     276,20,319,20,40,20,1517,50,379,50,
     47,50,0,10,0,10,0,10
2550 DATA 1432,20,358,20,45,20,1276,20,3
     19,20,40,20,1432,20,358,20,45,20,17
     03,50,426,50,53,50,0,10,0,10,0,10,1
     517,20,379,20,47,20,1276,20,319,20,
     40,20,1517,20,379,20,47,20,1911,50,
     478,50,60,50,0,20,0,20,0,20
2560 DATA 1517,50,379,50,47,50,1276,20,3
     19,20,40,20,1517,50,379,50,47,50,0,
     10,0,10,0,10,1517,50,379,50,47,50,1
     276,20,319,20,40,20,1517,50,379,50,
     47,50,0,10,0,10,0,10
2570 DATA 1432,20,358,20,45,20,1276,20,3
     19,20,40,20,1432,20,358,20,45,20,17
     03,50,426,50,53,50,1276,20,319,20,4
     0,20,1911,50,478,50,60,50,0,20,0,20
     ,0,20,999,999,999,999,999,999
2580 DATA 1276,100,319,100,159,100,1136,
     20,284,20,142,20,1073,50,268,50,134
     ,50,1276,20,319,20,159,20,0,100,0,1
     00,0,100
2590 DATA 1276,100,319,100,159,100,1136,
     20,284,20,142,20,1073,50,268,50,134
     ,50,1276,70,319,70,159,70,902,100,2
     25,100,113,100,999,999,999,999,999,
     999
2600 DATA 239,60,119,60,60,60,268,20,134
     ,20,67,20,284,20,142,20,71,20,319,2
     0,159,20,80,20,284,80,142,80,71,80
2610 DATA 478,9,239,9,119,9,0,1,0,1,0,1,
     478,10,239,10,119,10,358,19,179,19,
     89,19,0,1,0,1,0,1,358,19,179,19,89,
     19,0,1,0,1,0,1
2620 DATA 358,19,179,19,89,19,0,1,0,1,0,
     1,358,19,179,19,89,19,0,1,0,1,0,1,3
     79,20,190,20,95,20,358,20,179,20,89
     ,20,319,100,159,100,80,100
2630 DATA 478,9,239,9,119,9,0,1,0,1,0,1,
     478,9,239,9,119,9,0,1,0,1,0,1,478,2
     0,239,20,119,20,358,20,179,20,89,20
     ,319,20,159,20,80,20,284,40,142,40,
     71,40
2640 DATA 358,10,179,10,89,10,379,10,190
     ,10,95,10,426,20,213,20,106,20,268,
     19,134,19,67,19,0,1,0,1,0,1,268,19,
     134,19,67,19,0,1,0,1,0,1,268,40,134
     ,40,67,40
2650 DATA 239,10,119,10,60,10,268,10,134
     ,10,67,10,284,20,142,20,71,20,319,2
     0,159,20,80,20,358,20,179,20,89,20,
     379,20,190,20,95,20,358,20,179,20,8
     9,20,319,20,159,20,80,20,358,70,179
     ,70,89,70
2660 DATA 999,999,999,999,999,999
```

IT'S FOR YOO-HOO!



Join our Hero on a space and time odyssey through 53 screens, collecting crystals and staying alive

Roland IN Time

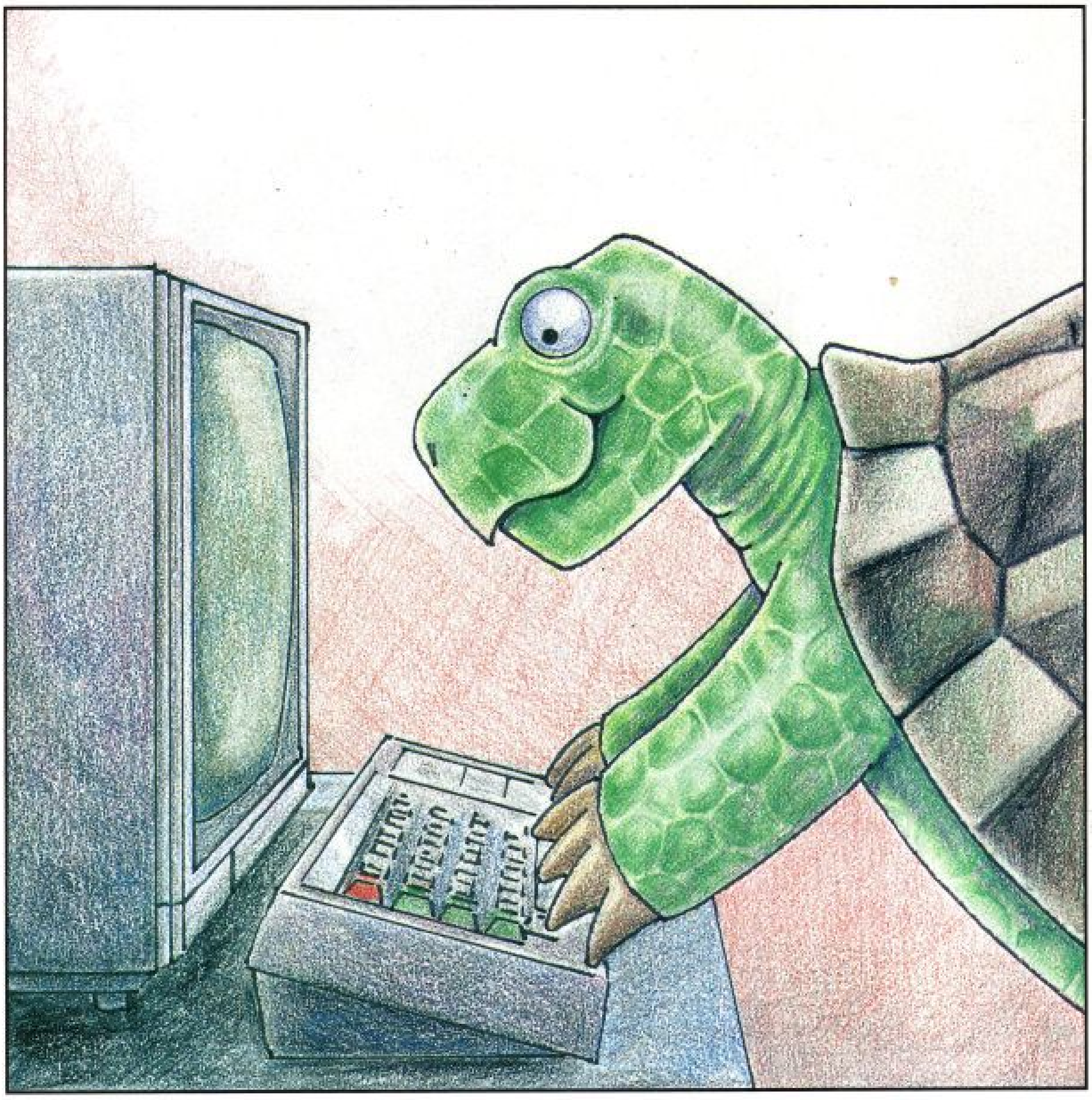
Roland in Time is for anyone with a CPC464!

SOFT 170

£8.95

AMSTRAD CPC464

Dr LOGO on the CPC464 : Boris Allan



Why have Amstrad decided to include Dr LOGO with the disc system they provide for the CPC464? What can you, the user, do with LOGO that you cannot perform in, for example, BASIC? Why should people without discs be anxious to get their hands on discs, so that they can use Dr LOGO?

Quite simply, LOGO is one of the most powerful languages available on microcomputers. The amazing thing about LOGO is that it can be used with ease by the very young, and yet LOGO is so flexible that it has been used successfully in the teaching of artificial intelligence (AI) to undergraduates. The range of applications for LOGO is extensive.

Interested in developing applications such as databases? LOGO could be the language for you, because quite complex and sophisticated sytems have been developed using LOGO, and LOGO has been programmed to produce an interpreter for the fifth generation language PROLOG. LOGO is being utilised in the teaching of history in schools, using LOGO to produce readily accessible databases from local records and documents.

Originally, LOGO was designed by the AI workers from the Massachusetts Institute of Technology (MIT): the idea behind LOGO was to produce a computer language environment which was easy to use, yet powerful enough to have the potential to develop complex applications. LOGO was aimed initially at school children, with the intention of helping the children in their use of language, by allowing easy manipulation of words and sentences. LOGO derived many of its ideas from the computer language LISP (see the user guide glossary about LISP).

One of the first LOGO 'applications' was a system to translate from English into a kind of 'pig Latin', that is, changing a sentence to another form which is not Latin but sounds as if it might be Latin. What happens is that the initial consonants are taken from the front of the word, and added to the end, together with the letters ay - this type of coded conversation is based on a form of forces slang. An example for the translation is the procedure which turns:

this is a sentence or so they tell me

into the pig Latin version

isthay isay aay entencesay oray osay eytthay elltay emay

and though the 'translation' is something which can be performed in BASIC, the programming does not come easily in that language.

You may have seen books which try to teach about artificial intelligence applications using BASIC, but that is like trying to listen to the pop group YES in mono (wearing a crash helmet?), rather than listening in stereo. YES still sound good in mono, but you need stereo for the full effect. Where AI is concerned, BASIC is like programming in mono, and you need LOGO (or LISP) to program in stereo - to get anywhere near the full effect.

Though LOGO was developed to help with children's use of language, LOGO is now more famous for the use of Turtle Graphics (the Glossary thinks so). Turtle graphics were invented by Prof Seymour Papert (of MIT) to provide an interesting focus for exploring programming. These graphics techniques have been so successful that they have been implemented in many other languages such as some ver-

sions of Pascal. The ideas behind turtle graphics are very simple.

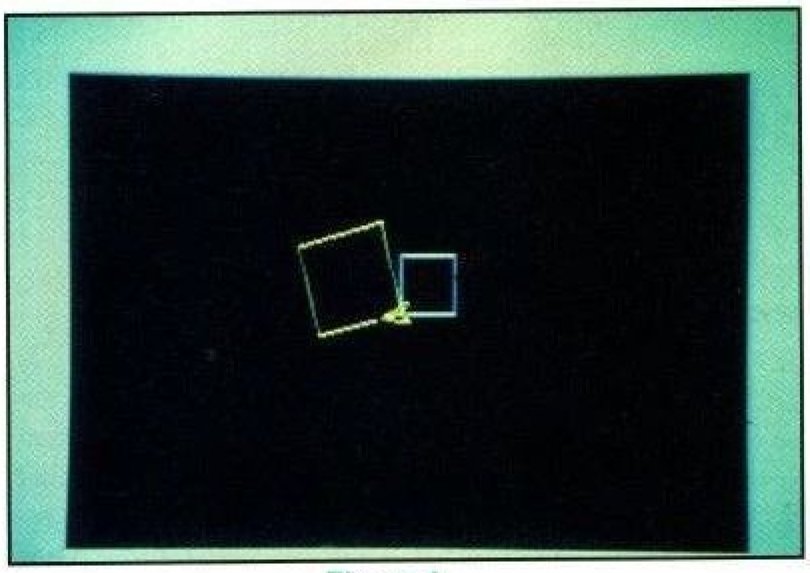


Figure 1

Think how you would give instructions to somebody to find their way: "Go on for three miles, turn right, carry on for another six miles and then turn left - it's two miles from the junction". In LOGO you draw shapes by providing instructions in this way, and the instructions you give are carried out by something called a 'turtle' Some turtles are devices which move around the flow (little controllable robots), but more often turtles are a design on the video screen. Though at least one version of LOGO uses a design which looks like a turtle, Dr LOGO uses a shape like an arrow head - which is more useful than a turtle shape, because you see more easily the direction in which the turtle is facing.

To get the turtle to draw a square on the screen, for example, we use the commands

```
fd 50 rt 90
fd 50 rt 90
fd 50 rt 90
fd 50 rt 90
```

and these commands are quite simple to understand. fd is short for forward, and rt is a short form of right. The turtle moves forward 50 units, turns right through 90 degrees, and the sequence is repeated three more times. If you want to check this out, pretend you are a turtle by walking out a square on the ground.

Another form of this set of instructions is:

```
repeat 4[fd 50 rt 90]
```

which seems self evident. If there is a variable (say) side, we can use the variable to direct how far forward we have to go (and thus the size of the side of the square):

```
repeat 4[fd :side rt 90]
```

where the colon in front of the variable name means that we are referring to the value of the variable. We can automate the drawing of squares by producing a procedure (which, with great originality, I will call square).

```
to square :side
repeat 4[fd :side rt 90]
end
```

where :side is an input (or parameter) which allows us to choose the size of the side of the square.

To draw a square of side 50, we have only to enter:

```
square 50
```

and a square of side 50 is drawn. The turtle is pointing in the same direction as at the start (check by going for a turtle walk), and so if we enter `rt 58` the turtle turns through 58 degrees to the right. To then enter:

```
square 100
```

draws another square (of side 100 units), at an angle to the first square.

Figure 1 shows the result of the sequence:

```
square 50 rt 58
square 100 lt 162
square 80
```

and if you look carefully you will be able to see the 'turtle' arrow. The turtle is the same colour as the last square to the drawn (that is, the square on the left), and you will note that each square is a different colour. The difference in colours is due to a subtle change in the square procedure. The new form is:

```
to square :side
setpc 1 + remainder
      (item 5 tf) 3
repeat 4 [fd :side rt 90]
end
```

where the line beginning `setpc` is new. Lines can be drawn in three different colours (plus the background colour), where the three colours are identified with the three 'pens' 1, 2, or 3 - the turtle uses one of three pens to draw its lines. the new line changes the pen colour each time the square is drawn in the sequence 1, 2, 3, 1, 2, ...

Item number 5 of the list of turtle facts (`tf`) is the present pen number, and the remainder is found when that number is divided by 3. The procedure remainder is defined by:

```
to remainder :p1 :p2
op :p1 - (int (:p1 /
   :p2) * :p2)
end
```

which outputs (`op`) the value shown. The procedure remainder is the equivalent of a function or operator in BASIC, because a funtion also produces an output, and in fact remainder corresponds to the BASIC operator MOD. Even with simple squares it is possible to produce intriguing effects, and Figures 2 and 3 are not more than uses of square.

In Figure 2 it is fairly easy to see how the squares are involved, and the creation of the effect was produced by:

```
to demo2 :side
cs fs ht
repeat 3 [square
     :side lt 120]
end
```

which is easy to understand, apart from `cs fs ht`. This line instructs "clear the screen, full screen for graphics, and hide the turtle". Figure 3 seems much more complex, but it is only built from squares, where the key line is:

```
square :side fd
       :side rt :angle
```

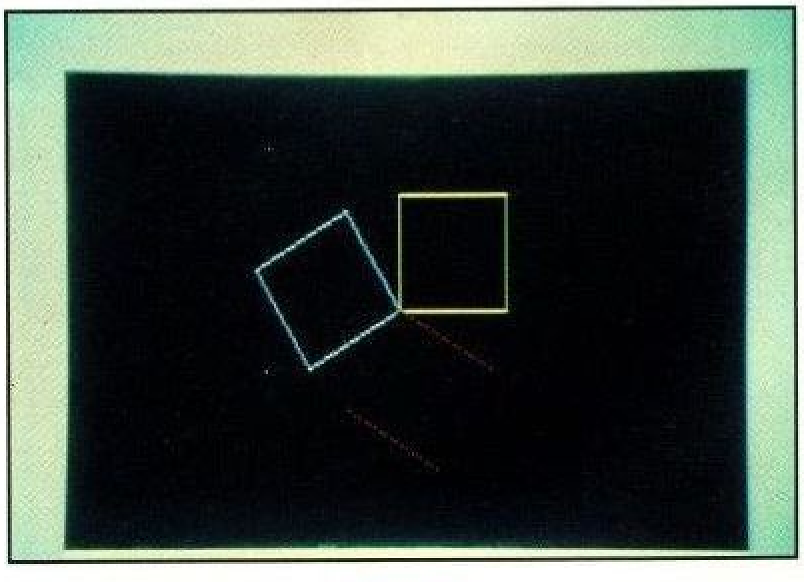


Figure 2

and, in Figure 3, the sides of successive squares grow by 3 units a time, and the :angle is 117.

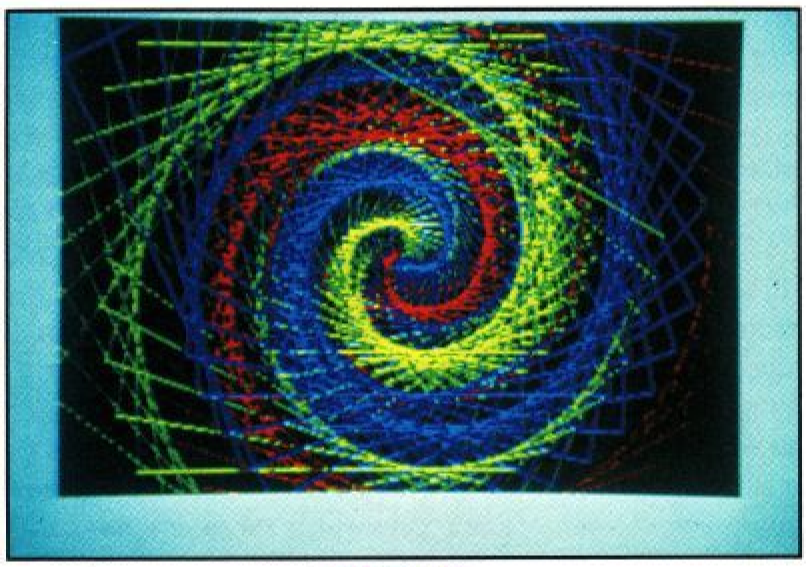


Figure 3

An important chracteristic of LOGO is shown in Figure 4, which is a wobble. To wobble you make a random turn, make a random move, choose a random pen colour, and then wobble again (which means you make a random turn, make a random move, choose a random colour, and then wobble again,...).

```
towobble
rt random 360 fd random
50setpc 1 + random 3
wobble
end
```

where, before the turtle wobbles, you enter:

wrap wobble

because this means that the lines can wrap round the edges of the screen, and will not disappear off the screen (as with the normal, or window, mode). The ability of a procedure to refer to itself is called 'recursion', and can be quite useful.

An intriguing example of recursion in action is the drawing of the windswept 'tree' in Figure 5. The procedure to draw the leaning three is called `ltree`, and I give this procedure so that those with disc systems can try it out (it also appears in the Guide to LOGO (SOFT160)). Do not worry if you cannot understand how the procedure works: it is only here for reference.

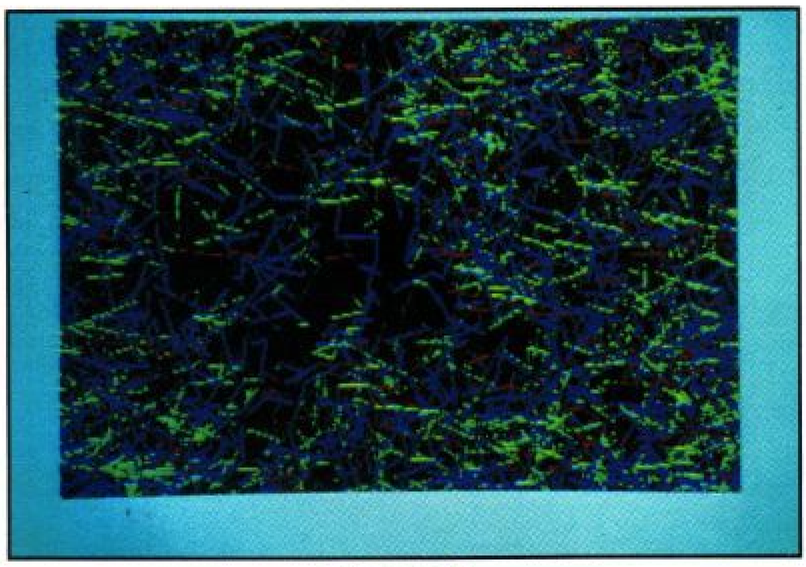


Figure 4

```
to ltree :length:angle :order :factor1 :factor2
ht pd
if :order = 0 [stop]
1t :angle fd :length
ltree :length * :factor1 :angle :order - 1
:factor1 :factor2
pu bk :length rt 2* :angle pd
fd :length *:factor1 :angle :order - 1
:factor1:factor2
pubk :length *:factor2 1t :angle pd
end
```

Figure 5 was produced by the call:

`ltree 90 35 9 0.8 0.4`

and, to help in seeing how the tree is formed, Figure 6 shows a rather more simple tree, produced by

`ltree 150 35 3 0.8 0.4`

A hint is that the tree starts from the bottom most point, and the drawing of the lines moves (with much retracing of the path) from the left side to the right.

When you are investigating Figure 5, remember that LOGO can produce not only complex graphical creations, but also equally complex numerical or verbal creations. I have used graphics as an example because it is easiest to 'see', but do not forget that LOGO is capable of far more.

---

1. See, for example, the Glossary to the CPC464 User Instructions concerning 'Fifth generation computers'.

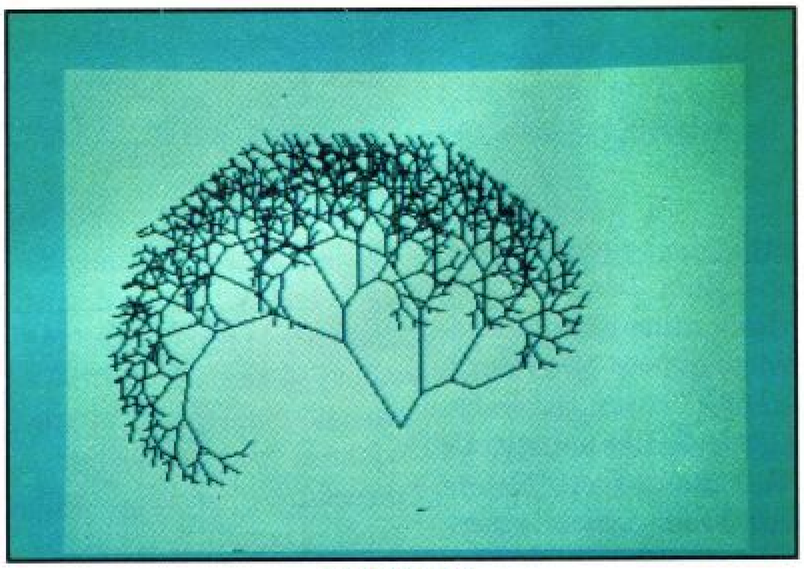
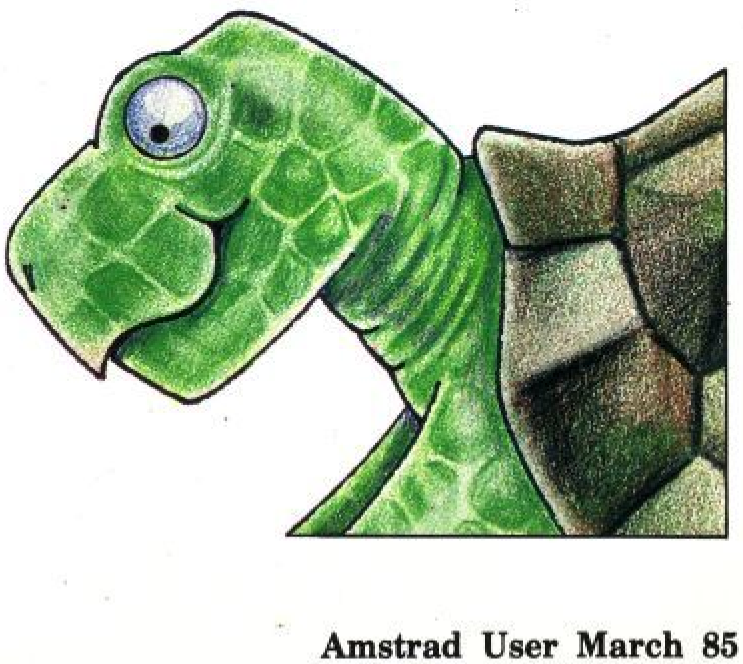


Figure 5



HARDWARE PRODUCTS from NORTHERN COMPUTERS

THE AMSTRAD CPC 464 COMPUTER IN EDUCATION



- **Outstanding hardware features include:**
  64K RAM memory, 32K ROM containing a high speed standard BASIC and the operating system, high quality Qwerty Keyboard with numeric keypad, built-in datacorder, high resolution graphics, 27 colours, 80 column text, 3 channel sound with stereo output and volume control, user port, parrallel printer interface.
- AMSTRAD disc drive system features include:
  *CP/M version 2.2, Dr Logo and AMSDOS which up-loads AMSOFT cassette programs to disc.

THE AMSTRAD EDUCATION SCHEME OFFERS:

(for bona fide education & training establishments only)

- **Substantial educational discounts.**
- **1 year FREE hardware service contract.**
- **Educational software lists for schools.**
- **Languages, utility, business and application software lists for higher education and training schemes.**
- **Future product information.**
- **Network development information.**

northern computers
micro computer systems for all applications

AMSTRAD'S EDUCATIONAL DISTRIBUTORS:

Tel: Frodsham (0928) 35700

Northern Computers Ltd., Churchfield Road, Frodsham, Cheshire WA6 6RD.

DUCKWORTH HOME COMPUTING

All books written by Peter Gerrard, former editor of *Commodore Computing International,* author of two top-selling adventure games for the Commodore 64, or by Kevin Bergin. Both are regular contributors to *Personal Computer News, Which Micro? and Software Review* and *Popular Computing Weekly.*

EXPLORING ADVENTURES ON THE AMSTRAD
by Peter Gerrard
£6.95

This is a complete look at the fabulous world of Adventure Games for the Amstrad Computer. Starting with an introduction to adventures, and their early history, it takes you gently through the basic programming necessary on the Amstrad before you can start writing your own games.
Inputting information, room mapping, movement, vocabulary – everything required to write an adventure game is explored in detail. There follow a number of adventure scenarios, just to get you started, and finally three complete listings written specially for the Amstrad, which will send you off into wonderful worlds where almost anything can happen.

The three games listed in this book are available on one cassette.

Other titles in the series include *Sprites & Sound on the 64, 12 Simple Electronic Projects for the VIC, Will You Still Love Me When I'm 64, Advanced Basic & Machine Code Programming on the VIC, Advanced Basic & Machine Code Programming on the 64,* as well as *Pocket Handbooks for the VIC, 64, Dragon, Spectrum* and *BBC Model B.*

Write in for a catalogue.



DUCKWORTH
The Old Piano Factory, 43 Gloucester Crescent, London NW1 7DY
Tel: 01-485 3484

★ NOW AVAILABLE FOR THE AMSTRAD CPC 464!!

WIN THE POOLS?

SPECTADRAW 3 – THE LATEST VERSION OF THE ORIGINAL AND BEST POOLS PREDICTION PROGRAM FOR THE 48K ZX SPECTRUM!!

AND NOW . . . AMSTRA-DRAW – THE FIRST POOLS PREDICTION PROGRAM FOR THE AMAZING NEW AMSTRAD CPC 464!!



- Supplied with Database containing data on over 10,000 matches since 1980!
- You update the Database each week – but no tedious typing, as team and division names already in program!
- Errors easily corrected – the program even checks your entries!
- Comprehensive instruction manual and menu driven program – easy to use, even for a newcomer to computing!
- Will forecast the least likely draws for those who prefer to bet on fixed odds!
- Built in perm generator – complete your coupon direct from the screen!
- Fully microdrive compatible! (Spectadraw only).
- Compatible with Currah Microspeech – the first pools program to read you its predictions! (Spectadraw only).

Spectadraw 3 for the 48K Spectrum .... £9.95 inclusive
Amstra-Draw for the Amstrad CPC 464 .... £9.95 inclusive
(Cheques/P.O.s payable to B. S. McALLEY)

We dispatch every Monday with the database made up to include all matches up to the date of dispatch.

SPECTADRAW (Dept CU), 1 Cowleaze, Chinnor, Oxford OX9 4TD.
(Tel: 0844-52426)

Amstrad User Software Directory.

This is a list of all the software we know about, the information has been collated from various sources and is as accurate as possible. However Amstrad User does not claim to be perfect and cannot be held responsible for any inaccuracies or omissions. It would be wise to check before sending any money.

If you run a software house please send two copies of your products to the magazine so that we can review the program and include it in the directory.

Alligata Software Ltd, 1 Orange Street, Sheffield,S1 4DW,	Blagger	Ladders and platforms game.	£7.95
0742 755796	Defend or Die	Defender type alien Zap.	£7.95
Anirog Software Ltd, 29 West Hill, Dartford, Kent, DA1 2EL,0332 92513	Flight Path 737	Flight simulator	£6.95
	3D Time Trek	Arcade game	£7.95
	Zodiac	Arcade Game	£6.95
	House of Usher	Arcade game	£7.95
	Survivor	Monster and maze game	£7.95
	Moon Buggy	Lunar journey amist aliens and a rocky landscape	£7.95
Applied Systems Knowledge Ltd.,London House, 68 Upper Richmond Road, London,SW15 2RP, 01-874-6046	Number Painter	Educational. Mental Arithmetic for ages 5 to 14	£8.95
Argus Press Software,Liberty House,222 Regent Street,London W1,01-437-0626	American Football	Tactical strategy game	£9.95
	Music 7+	Educational music tutor	£8.95
	Party time	Educational	£8.95
	CPC464 Computing	Bi-monthly tape magazine	£3.99
CDS, Silver House, Silver Street, Doncaster, South Yorkshire,BN1 1HL	Steve Davis Snooker	Accurate Snooker Game	£7.95
Cable Software,52 Limebury Road,Luton,Beds.	Trace Race	Light cycles type game	£6.95
Camel Micros, 33a Commercial Road, Weymouth, Dorset	Grasp primer	Graph drawing uttility	£8.50
	Toolbox II	Extra commands for programmers	£4.95
	Number 11	Utility	£5.50
Cases Computer Simulations,14 Langton Way,London SE3 7TL	Warzone	Strategy Game	£6.95
CP Software,2 Glebe Road,Uxbridge,Middx,UB8 2RD	Bridge Player	Card game	£9.95
	Superchess	Chess program	£9.95
CRL Group PLC. 9 Kings Yard,Carpenters Road,London E15 2HD	Test Match	Cricket Simulation	£6.95
	Glug Glug	Arcade Game	TBA
	AHHH!	Arcade Game	TBA
	War of the Worlds	Graphic adventure of the record of the film of the book	TBA
David Computer Software,38 South Parade,Bramhall,Stockport	Maths and Physics	O-level tutor	£6.00
	Sound Effects		£5.00
	Printer package	Typewriter facility on screen	£6.00
	Arnold Answers Back	Conversation Program	£5.00
	Agent 007		£5.00
	Beanfeast	Version of the 'beans' board game	£5.00
	Stock Control		£6.00
	Report Writer		£8.00
	Data Bank		£10.00
	Invoice Control		£6.00
	Club Secretary		£10.00
	Sundry Creditors		£6.00
Dean Software Ltd., P.O. BOX 1, The Mews, Mitcheldean, Glos. GL17 0HH	Easidata	Flexible database program	£19.99
	Easivat	VAT book-keeping program	£39.99
	Dean database/Mailing list	Electronic card index	£12.99
	Music composer	Graphic music editor	£9.99

Design Design, 2 Ashton Way, East Herrington, Sunderland, SR3 3RX	Dark Star	Fast, spectacular, 3D space game	£7.95
Eborsoft,81 Hull Road, York YO1 3SS	Dice Games	Simple dice program	£11.00
English Software Company,PO Box 43,Manchester,M60 3AD,061-835-1358	Jet Boot Jack	Arcade Game	£7.95
Gemini Marketing Ltd., 18a Littleham Road, Exmouth, Devon, EX8 2QG	Database		£19.95
	Report Generator	Businessmans Tool	£19.95
	Home Accounts		£19.95
Hewson Consultants,56B Milton Trading Estate, Milton, Abingdon, Oxon, OX14 4RX	Technician Ted	Ladders and platforms game	£7.95
	Heathrow International ATC	Airport Simulator	TBA
Hisoft, 180 High Street North, Dunstable, Beds. LU6 1AT, 05 82 696421	Pascal	High level, compiled language	£24.95
	Devpac	Machine code development tool	£24.95
	Font 464	Character design utillity	£7.95
Interceptor Software,Lindon House,The Green, Tadley, Hants.	Forest at Worlds End	Graphic Adventure	£6.00
	Heroes of Karn	Graphic Adventure	£6.00
	Jewels of Babylon	Graphic Adventure	£6.00
	Message from Andromeda	Graphic Space Adventure	£6.00
Kuma Computers Ltd. 12 Horseshoe Park, Horseshoe Road, Pangborne, Berks RG8 7JW	Holdfast	Dictatorship game	£5.95
	Gems of Stradus	Adventure game	£7.95
	Star Avenger	"Scramble" type game	£6.95
	Galaxia	Sideways scrolling space game	5.95
	Music Composer	Utility	£9.95
	LOGO	High level language	£19.95
	Database	Electronic card index	£14.95
	Zen Assembler	Machine code development tool	£19.95
	Easivat	VAT book-keeping program	£39.50
	Home Budget	Home finance program	£19.95
	Fruity Frank	Game	£6.95
Level 9 Computing, 229 Hughenden Road, High Wycombe, Bucks,HP13 5PG	Colossal Adventure	Huge text adventure	£9.95
	Lords of time	Text adventure	£9.95
	Adventure Quest	text adventure	£9.95
	Dungeon Adventure	Text adventure	£9.95
	Snowball	Space adventure	£9.95
	Return to Eden	Very difficult adventure with graphics	£9.95
	Emerald Isle	Adventure with graphics	£6.95
Llamasoft, 49 Mount Pleasent,Tadley,Hants,RG26 6BM	Ancipital	Alien Zap adventure	TBA
	Psycadelia	Light synthesiser	TBA
Microbyte Software, 18 Hilgrove Road, Newquay, Cornwall, TR7 2QZ	ER*BERT	"Q*Bert" style game	£5.95
Micropower Ltd. Northwood House, North Street, Leeds,LS7 2AA	Ghouls	Tricky ladders and platforms game	£6.95
Mikro-Gen, 44 The Broadway, Bracknell, 0344 427317	Pyjamarama	Ladders and platforms type game	£8.95
Melborne House Publishers,Castle Yard House, Castle Yard, Richmond TW10 6TF	Sir Lancelot	Ladders and platforms type game	£6.95
	The Hobbit	The adventure game of the book	£14.95
Nemisis, 10 Carlow Road, Ringnstead, Ketttering, Northhants, NN14 4DW	The Trial of Arnold Blackwood	Fun adventure	£6.50
	Arnold goes to somewhere else!	Adventure	£6.50
	The Wise and Fool of Arnold Blackwood	Adventure	£6.50
	New Angelique: The Grief Encounter	Romantic adventure	£6.50
Pride Utilities, Dept C4, 7 Chalton Hts., Chalton, Luton, Beds.	Zedis	Editor and disassembler - machine code utillity	£6.95

Skywave Software, 73 Curzon Road, Bornemouth, BH1 4PW, 0202-302385	Forth	Compiled language	POA
Spectradraw, Dept CU,1 Cowleaze, Chinnor, Oxford, OX9 4TD	Amstra-draw	Pools prediction program	£9.95
Tasman software, Springfield House, Hyde Terrace, Leeds LS2 9LN	Tasword 464	Word processor	£19.95
	Tasprint 464	Variable fonts for dot-matrix printers	£9.90
	Tascopy 464	High resolution screen dumps with shading	£10.90
Timatic systems Ltd., Fareham Market, Fareham, Hants, 0329 236727		Suppliers of CP/M programs to order	POA
Timeslip software, Stoneybarn workshops, The old primary school, Main Street, Stoneyburn, West Lothian, Scotland, EH47 8AP	Rollaball	Game	£6.95
	The Royal Quest	Adventure	£6.95
	The Moors Challenge	Board game	£6.95
Virgin Games,2-4 Vernon Yard, Portabello Road, London W11 2DX	Sorcery	Ladders and platforms type game	£8.95
Wicca-soft, 105 Bollington Road, Bollington, Macclesfield, Cheshire, SK10 5EL	Crystal Theft	Text adventure	£7.99
All the programs listed below are avaliable from Amsoft at 169 Kings Road,Brentwood, Essex, CM14 4EF. 0277 230222			
Abersoft	Classic Adventure	A version of the first ever adventure	£8.95
Alligata	Blagger	Ladders and platforms game	£8.95
Argus Press Software	American Football	Strategy Game	£9.95
	Detective	Whodunnit elimination game for 2 to 6 players	£8.95
	Stockmarket	Trading game	£8.95
Artic Computing	Mr Wongs Loopy Laundry	Ladders and platforms game	£8.95
	Mutant Monty	Arcade/Maze Game	£8.95
Ian Beynon	Astro attack	Alien filled maze game	£8.95
Borne Educational Software	Wordhang	Hangman with 250 words	£8.95
	World-wise	Geography program	£8.95
	Map Rally	Educational program	£8.95
	Animal, Vegetable, Mineral	educational program	£8.95
	Happy Writing	Writing program for 3-6 year olds	£8.95
	Happy letters	educational software for 3-6 year olds	£8.95
	Happy Numbers	/	£8.95
	Timeman One	Tell the time tutorial	£8.95
	Timeman Two	Educational 24 Hour Clock	£8.95
Britannia Software	Grand Prix Driver		£8.95
	Home Runner		£8.95
Computersmith	Ams-golf	Golf game	£8.95
	Roland Ahoy!	Tactical Arcade game	£8.95
Dean Software	Xanagrams	Word game	£8.95
Durrell	Harrier Attack	Arcade Game	£8.95
	Space Hawks	Galaxians-type game	£8.95
Epicsoft	Bridge-It	arcade game	£8.95
	Roland on the Run	Frogger-type game	£8.95
	Centre Court	Tennis game	£8.95
Gem Software	Roland Goes digging	Monster trapping maze game	£8.95
	Oh Mummy	Maze game	£8.95
	Spannerman	Plumbers nightmare maze game	£8.95
	Snooker	Snooker simulation	£8.95
	Roland in Time	Ladders and platforms game	£8.95
	Sultans Maze	3D Maze game with monsters	£8.95
Incentive Software	Splat	tricky sliding maze game	£8.95
Indescomp	Roland in the caves	Flea caught in a cave, arcade game	£8.95
Indescomp	Roland on the ropes	Scrolling maze game	£8.95
	Galactic Plague	Galaxians type game	£8.95

Kuma Computers	Gems of Stradus	Simple graphics adventure	£8.95
Ocean	Hunchback	Arcade game	£8.95
Protek	Hunter killer	Submarine game	£8.95
Micro-mega Software	Haunted Hedges	Maze Game	£8.95
	Codename mat	Space pilot game	£8.95
Mikro-Gen	Master Chess	Chess program	£8.95
	Laserwarp	Alien shoot-out arcade game	£8.95
Mr Micro	Cubit	3D four in a row	£8.95
	Punchy	Hunchback type game	£8.95
	Crazy Golf		£8.95
Romik Software	Alien Break-in	Arcade Game	£8.95
Paul Shirley	Quack a Jack	Arcade/maze game	£8.95
Software Projects	Manic Miner	The game that launched a thousand clones	£8.95
Softspot	Electro Freddy		£8.95
Terminal Software	Star Commando	Alien shooting game	£8.95
Temptation Software	Admiral Graf Spee	Strategy Game	£8.95
Intelligence of Ireland	Microscript	CP/M Word processing program	£49.95
	Micropen	CP/M database program	£49.95
Kuma	Home budget	Domestic finance program	£19.95
Saxon Computing	Easi-Amscalc	Spreadsheet	£19.95
Juniper Computing	Easi-Amsword	Simple wordprocessor	£9.95
Tasman Software	Amsword	Word processor	£19.95
DJL Software	Screen designer	Utillity	TBA
Hisoft	Devpac	Machine code development tool	£24.95
	Hisoft Pascal	High level language	£34.95
Amsoft	Pitman typing tutor		£9.95

Do you Belong?

All the best people are in the Amstrad User Club.

For £19.95 you get :-

* Discount and privilege card.
* Latest news on Amsoft products.
* Free entry to Amstrad's £2000 competition.
* One year subscription to Amstrad User.
* Binder and divider for Amsoft documentation supplements.

Send today to

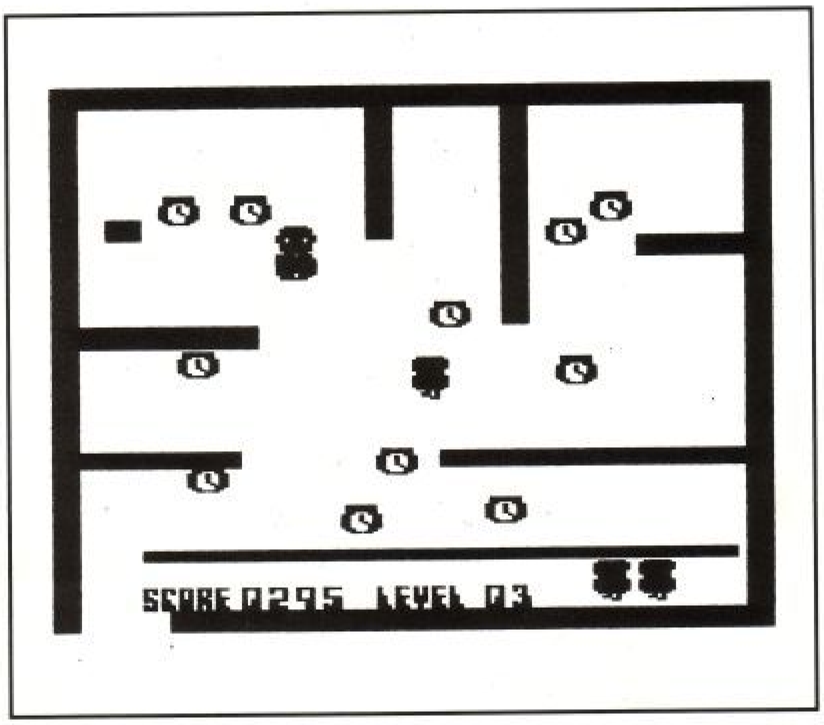
Amsoft, Brentwood House, 169 Kings Road, Brentwood, Essex CM14 4EF

Events & Screen Dumps

By use of the CPC464s in-built event mechanism it is quite possible to run more than one program at the same time. In other words, MULTI - TASKING, as provided on other, more expensive machines, is also available on your Arnold.

A second task may, for example, be loaded and set running before another program is loaded then if this second task finds a particular key combination pressed while the program is running it could dump the current contents of the screen to a printer - thus enabling any screen from a program to be printed out (though a necessarily monochrome dump of a multi-coloured screen may lose something in the translation). The program here does just this. It could also be used to dump a screen within a more serious program such as Masterfile, though in all cases, care should be taken to make sure that the program is not loaded at an address that would be in contention with the main program. Finally, it could also be loaded and run along with your own programs to dump the screen of, for example a graph drawing program.

The machine code program described in this article is written in three distinct sections, the first comprises a relocator as introduced in last months article on RSXs, the second sets up an 'event routine' and the final section is the actual printer dump routine. These three sections may be employed in your own programs quite separately - the printer



dump could for instance be assembled on its own if memory space were at a premium, however you would lose the advantage of it being relocatable. Similarly, the code to set up an event routine may be modified for a specific application. Finally, the relocator may be used on any program, as shown by last months RSX example.

No further details will be given here of the relocator, as this has previously been described, however, the event mechanism warrants further description. The CPC464 relies quite heavily on interrupts for its standard operation, however these are usually dealt with at a higher level using the event mechanism. The firmware provides several routines to construct 'event blocks' and add them to one of several types of 'kicking' mechanism.

The most basic form of an event block consists of a seven byte area of memory containing such information as a link to other blocks, a count of how many times the block has been 'kicked' (more of this later) and the address of the event routine. A firmware routine

KL_INIT_EVENT #BCEF

is provided that will set up an event block which could then be passed to one of the routines to set up a tick block (a rechargeable count down timer). An event may be set up to be of one of several classes of event, synchronous/asynchronous, express or normal. Explanation of the particular event mechanism used in this example may help to make the method by which events are used a little clearer.

The program makes a check every 1/50th of a second to see if the COPY key has been pressed with CTRL and if so a call is made to the routine that dumps the contents of the screen. The event mechanism employed is a normal asynchronous event that is kicked by the frame flyback event. That is, every 1/50th of a second the frame flyback occurs and increments a count within the event block that has previously been initialised by

KL_NEW_FRAME_FLY

At the end of the machines own interrupt routine, it checks the counts within any currently active blocks and decrements the count in any which require service. It then performs the corresponding routine whose address is contained within the event block. In this example, this performs a short piece of code that checks for a specific key combination being pressed, if the two keys are not pressed then processing returns to the machines own housekeeping (probably running the BASIC). If they are, then the screen dump routine is entered. The result of this is that the key checking routine is run transparently to the currently running program. In effect a second, background task is running alongside the foreground program - true multi-tasking.

There are several ways in which one can arrange to have an event block 'kicked' (i.e. have its service count incremented so that it is later processed). This is achieved by adding the event block onto a particular kicking mechanism, this can be either the frame flyback event (every 1/50th second),

KL_NEW_FRAME_FLY #BCD7

the fast ticker event (every 1/300th second),

KL_NEW_FAST_TICKER #BCE0

or the normal ticker event (every 1/50th second),

KL_ADD_TICKER #BCE9

Each of these requires the address of a corresponding event/ticker block to be passed to it. Full details of each of these is given in the Firmware Manual, Appendix 10 and Chapters 10 and 11. The frame flyback, used in this example, is probably the simplest to use as it requires only a couple of extra bytes on the front of a standard event block, all of which is easily set up by a call to

KL_NEW_FRAME_FLY.

This routine takes the address of the event block in HL, the ROM select address in C (set to zero if a near address is provided), the actual address in DE and the event class in B. The event class is a bit significant number whose meaning is given below:

Bit 0 : If 1 then address is near - i.e. C (ROM select) is ignored

Bit 1 to Bit 4: priority number if event is synchronous

Bit 5 : Always zero

Bit 6 : If 1 then event is express

Bit 7 : If 1 then event is asynchronous

In the example, this is %10000001, an asynchronous event with a near address. A synchronous event is serviced when the main program is ready for it to be serviced, it would then call

KL_NEXT_SYNC

to determine if any events are outstanding on the queue and if so it may then call

KL_DO_SYNC

to process it. An asynchronous event, on the other hand, will be processed when the machine comes to the end of its standard interrupt path, if it has been kicked recently. The exact time at which such an event will be run is therefore indeterminate. In this example the exact time at which the keys are checked does not matter.

Another example application for the event mechanism would be to intercept the firmware jump block entry that is used to send characters to the printer and send them, instead, to a buffer in memory. Then a second task could be run which constantly checks this buffer and if it finds anything there it may direct these to the printer - in effect a printer spooler. This would enable one to type LIST #8 and a few moments later resume BASIC programming while the program was actually being printed concurrently.

The printer dump routine in this example uses a six byte data area that is indexed using the IX register. The usage of each byte is as follows:

IX+0 : byte that is built up and then sent to the printer

IX+1 and IX+2 : low and high bytes of the X co-ordinate

IX+3 and IX+4 : low and high bytes of the Y co-ordinate

IX+5 : lines printed so far

IX+6 : bits built up so far

The program starts by initialising X , Y to (0,400) and the other pieces of data to zero. The main loop then starts by sending an escape sequence to the printer that instructs it to interpret the following bytes as graphic dot data. After the byte out and bit count have been set to zero an inner loop is entered that TESTs the 7 pixels below the current position and SETs and shifts the byte that is being built up. When all 7 bits have been picked up, the byte is printed. X is moved on by 2, then a check is made to see if it is 642 yet. If not, then Y is moved back up by 7 pixels and the loop returns. If X has reached the end of the line, the line count is incremented and a LF/CR sequence is sent to the printer, X is then set to zero, finally the line count is checked to see if 34 lines have been printed, if not then the main loop returns, otherwise the printer is sent a character 15 to set it back to normal text printing mode and the routine ends. At the end of each line a check is made to see if the SHIFT key is pressed, if so the printer dump ends - this enables one to abort the dump without having to wait for the whole screen to be printed.

Listing 1 is the assembler source code that may be used with Hisofts DEVPAC and could be modified to your own requirements. Listing 2 is a standard machine code loader that may be used to input the hex values that are given in listing 3. Note that the final byte on each line is not a byte of data but a checksum for that line and may be ignored if you wish to use the data with some other hex loading program. Note also that it is not necessary to type in all the instructions, just ignore lines 270 onwards and delete line 50. When asked for a load address, &8000 will do, then, when all the data has been entered it can be saved using:

```
SAVE"grabscrn.bin",b,&8000,&140
```

To use the program at a later date one can set HIMEM down by 320 bytes, then load the code at the new HIMEM value and CALL it.

```
10 MEMORY HIMEM-321
20 hm=HIMEM+1
30 LOAD"grabscrn.bin",hm
   :CLOSEIN
40 CALL hm
50 NEW
```

When RUN, this program moves HIMEM down to accomodate the block of code. The variable hm is set to HIMEM+1, rather than using `LOAD"...",HIMEM+1` because the point at which HIMEM is moved during a LOAD command is indeterminate. The CLOSEIN forces the cassette buffer memory to be returned for BASICs use. The CALL hm executes the code which relocates itself then starts the second task running. From this point on, pressing CTRL/COPY causes the current screen contents to be dumped to the printer. One could now load a program and, as long as it does not use the same memory at which the code has been loaded or reset the event mechanism in some way, then it will be possible to print any screen from it. To convert this program so that it will work with an EPSON compatible (spelling) the following additions and changes should be made:

Add

```
801 LD A,#1B
802 CALL MC_PRINT_CHAR
803 LD A,#41
804 CALL MC_PRINT_CHAR
805 LD A,#06
806 CALL MC_PRINT_CHAR
```

these lines set the printers line feed of the printer to 6/72 inch. Change the numbers in lines 870 and 890 to #40 and #01 respectively (note that these appear to be in the opposite order to the previous values, i.e. lsb,msb rather than msb,lsb - this is correct). Change the operation in line 940 from **SRA** to **SLA**. Also, change the operation in 1020 from SET 6,(IX+0) to SET 0,(IX+0). For those not lucky enough to own the DEVPAC assembler, the modified hex dump is as shown in listing 4.

```
Hisoft GENA3.1 Assembler. Page     1.
Pass 1 errors: 00
8000              13            org   #8000
8000              20 entry:     equ   $
8000  21E1E9      30            ld    hl,#e9e1            ; start RELOCATOR
8003  223000      40            ld    (#30),hl
8006  F7          50            rst   #30
8007  EB          60 this:      ex    de,hl
8008  213001      70            ld    hl,rel_tab-this
800B  19          80            add   hl,de          ; HL has abs addr of rel_table
800C  4E          90 loop:      ld    c,(hl)
800D  23         100            inc   hl
800E  46         110            ld    b,(hl)         ; BC has first entry in table
800F  79         120            ld    a,c
8010  B0         130            or    b
8011  2815       140            jr    z,done         ; entry was 0000 so exit loop
8013  E5         150            push  hl             ; stack current table pointer
8014  60         160            ld    h,b
8015  69         170            ld    l,c
8016  19         180            add   hl,de          ; make table entry absolute
8017  E5         190            push  hl             ; and stack this value
8018  4E         200            ld    c,(hl)
8019  23         210            inc   hl
801A  46         220            ld    b,(hl)     ; get contents pointed to by entry
801B  60         230            ld    h,b
801C  69         240            ld    l,c
801D  19         250            add   hl,de                ; make this absolute
801E  44         260            ld    b,h
801F  4D         270            ld    c,l
8020  E1         280            pop   hl           ; retrieve absolute entry value
8021  71         290            ld    (hl),c
8022  23         300            inc   hl
8023  70         310            ld    (hl),b   ; store absolute value at this addr.
8024  E1         320            pop   hl         ; get addr of previous table entry
8025  23         330            inc   hl             ; make HL point at next
8026  18E4       340            jr    loop
8028             350 done:      equ   $              ; end of RELOCATOR
                 360 ;
BCD7             370 kl_new:    equ   #bcd7
BB1E             380 km_tes:    equ   #bb1e
0009             390 COPY_k:    equ   9
0017             400 CTRL_k:    equ   23
0015             410 SHIFT_:    equ   21
8028  214900     420 r01:       ld    hl,tick_block-this   ; set up EVENT
802B  0681       430            ld    b,%10000001
802D  0E00       440            ld    c,0
802F  112F00     450 r02:       ld    de,int_routine-this
8032  CDD7BC     460            call  kl_new_frame_fly
8035  C9         470            ret
                 480 ;
8036             490 int_ro:    equ   $           ; this is the interrupt routine
8036  C5         500            push  bc
8037  D5         510            push  de
8038  E5         520            push  hl
8039  F5         530            push  af
803A  3E09       540            ld    a,COPY_key
803C  CD1EBB     550            call  km_test_key
803F  280A       560            jr    z,return
8041  3E17       570            ld    a,CTRL_key
8043  CD1EBB     580            call  km_test_key
8046  2803       590            jr    z,return
8048  CD5900     600 r03:       call  scrndump-this
804B  F1         610 return:    pop   af
804C  E1         620            pop   hl
804D  D1         630            pop   de
804E  C1         640            pop   bc
804F  C9         650            ret
                 660 ;
8050             670 tick_b:    defs  9
8059             680 FRED:      defs  7
8060             690 scrndu:    equ   $              ; start of PRINTER dump
BBF0             700 GRA_TE:    EQU   #BBF0
BD2B             710 MC_PRI:    EQU   #BD2B
BD2E             720 MC_BUS:    EQU   #BD2E
                 730 ; DMP1DUMP
8060  DD215200   740 r04:       LD    IX,FRED-this
8064  DD360100   750            LD    (IX+1),0        ; X co-ord. = 0
8068  DD360200   760            LD    (IX+2),0
806C  DD360390   770            LD    (IX+3),#90      ; Y co-ord. = 400
8070  DD360401   780            LD    (IX+4),#01
8074  DD360500   790            LD    (IX+5),0        ; line count = 0
8078  DD360000   800            LD    (IX+0),0        ; byte out = 0
807C  CD2EBD     810 MAIN:      CALL  MC_BUSY_PRINTER
807F  38FB       820            JR    C,MAIN
8081  3E1B       830            LD    A,#1B
8083  CD2BBD     840            CALL  MC_PRINT_CHAR       ; ESC
8086  3E4B       850            LD    A,#4B
8088  CD2BBD     860            CALL  MC_PRINT_CHAR       ; "K"
808B  3E02       870            LD    A,#02
808D  CD2BBD     880            CALL  MC_PRINT_CHAR
8090  3E40       890            LD    A,#40
8092  CD2BBD     900            CALL  MC_PRINT_CHAR       ; #240
8095             910 LINE:      EQU   $
8095  DD360000   920            LD    (IX+0),#00          ; byte out = 0
8099  DD360600   930            LD    (IX+6),#00          ; 7 bit count = 0
809D  DDCB002E   940 BYTE:      SRA   (IX+0)
80A1  DD5E01     950            LD    E,(IX+1)            ; FRED+1-2 = X co-ord.
80A4  DD5602     960            LD    D,(IX+2)
80A7  DD6E03     970            LD    L,(IX+3)            ; FRED+3-4 = Y co-ord.
80AA  DD6604     980            LD    H,(IX+4)
80AD  CDF0BB     990            CALL  GRA_TEST_ABSOLUTE
80B0  B7        1000            OR    A
80B1  2804      1010            JR    Z,NOSET
80B3  DDCB00F6  1020            SET   6,(IX+0)       ; +0 is byte being built
80B7  DD3406    1030 NOSET:     INC   (IX+6)         ; increase bit count
80BA  DD7E06    1040            LD    A,(IX+6)       ; and get it
80BD  FE07      1050            CP    #7
80BF  2812      1060            JR    Z,PRINT        ; if got 7 bits then print
80C1  DD6E03    1070            LD    L,(IX+3)       ; get Y co-ord.
80C4  DD6604    1080            LD    H,(IX+4)
80C7  2B        1090            DEC   HL           ; use HL for simple 16 bit maths
80C8  2B        1100            DEC   HL             ; move down by 2 (1 pixel)
80C9  DD7503    1110            LD    (IX+3),L       ; then put it back
80CC  DD7404    1120            LD    (IX+4),H
80CF  18CC      1130            JR    BYTE
80D1  18A9      1140 MID_JM:    JR    MAIN         ; to relative jump more than 128
80D3  CD2EBD    1150 PRINT:     CALL  MC_BUSY_PRINTER
80D6  38FB      1160            JR    C,PRINT        ; wait until its free
80D8  DD7E00    1170            LD    A,(IX+0)       ; get byte to output
80DB  CD2BBD    1180            CALL  MC_PRINT_CHAR  ; and print it
80DE  DD3401    1190            INC   (IX+1)
80E1  DD3401    1200            INC   (IX+1)         ; move X on by 2
80E4  2003      1210            JR    NZ,NOHIGH
80E6  DD3402    1220            INC   (IX+2)         ; handle high byte if necess.
80E9  DD7E01    1230 NOHIGH:    LD    A,(IX+1)
80EC  D682      1240            SUB   #82
80EE  2007      1250            JR    NZ,RESETY
80F0  DD7E02    1260            LD    A,(IX+2)
80F3  FE02      1270            CP    #02
80F5  2812      1280            JR    Z,ENDLIN            ; X = #282
80F7  DD6E03    1290 RESETY:    LD    L,(IX+3)
80FA  DD6604    1300            LD    H,(IX+4)
80FD  110C00    1310            LD    DE,#000C
8100  19        1320            ADD   HL,DE
8101  DD7503    1330            LD    (IX+3),L            ; increase Y by 12
8104  DD7404    1340            LD    (IX+4),H
8107  188C      1350            JR    LINE
8109  DD3405    1360 ENDLIN:    INC   (IX+5)              ; increment line count
810C  3E0A      1370            LD    A,#0A               ; Line Feed
810E  CD2BBD    1380            CALL  MC_PRINT_CHAR
8111  3E0D      1390            LD    A,#0D               ; Carriage Return
8113  CD2BBD    1400            CALL  MC_PRINT_CHAR
8116  3E15      1410            LD    A,SHIFT_key
8118  CD1EBB    1420            CALL  km_test_key     ; check if SHIFT is pressed
811B  200F      1430            JR    NZ,BYE
811D  DD360100  1440            LD    (IX+1),0            ; set X to zero
8121  DD360200  1450            LD    (IX+2),0
8125  DD7E05    1460            LD    A,(IX+5)            ; get line count
8128  FE22      1470            CP    #22
812A  20A5      1480            JR    NZ,MID_JMP         ; have we done 34 lines (7 high)
812C            1490 BYE:       EQU   $
812C  3E0F      1500            LD    A,15
812E  CD2BBD    1510            CALL  MC_PRINT_CHAR
8131  3E0D      1520            LD    A,13
8133  CD2BBD    1530            CALL  MC_PRINT_CHAR
8136  C9        1540            RET
8137            1550 rel_ta:    equ   $
8137  2200      1560            defw  r01-this+1
8139  2900      1570            defw  r02-this+1
813B  4200      1580            defw  r03-this+1
813D  5B00      1590            defw  r04-this+2       ; addr in IX op is 3 and 4
813F  0000      1600            defw  0
```

DISCOVER THE DISC DRIVE AT COMET ... NOW!

You can now get the Amstrad DDI-1 disc drive at Comet . . . complete with interface, CP/M* **(giving you the option to access 3000 proven programmes)** and LOGO with its famous educational applications.

It's a system that helps you get a great deal more from a great computer.

Why? Because the disc gives you much faster access to programmes.

For example, a loading time of seconds compared with the minutes taken by a cassette containing the same data.

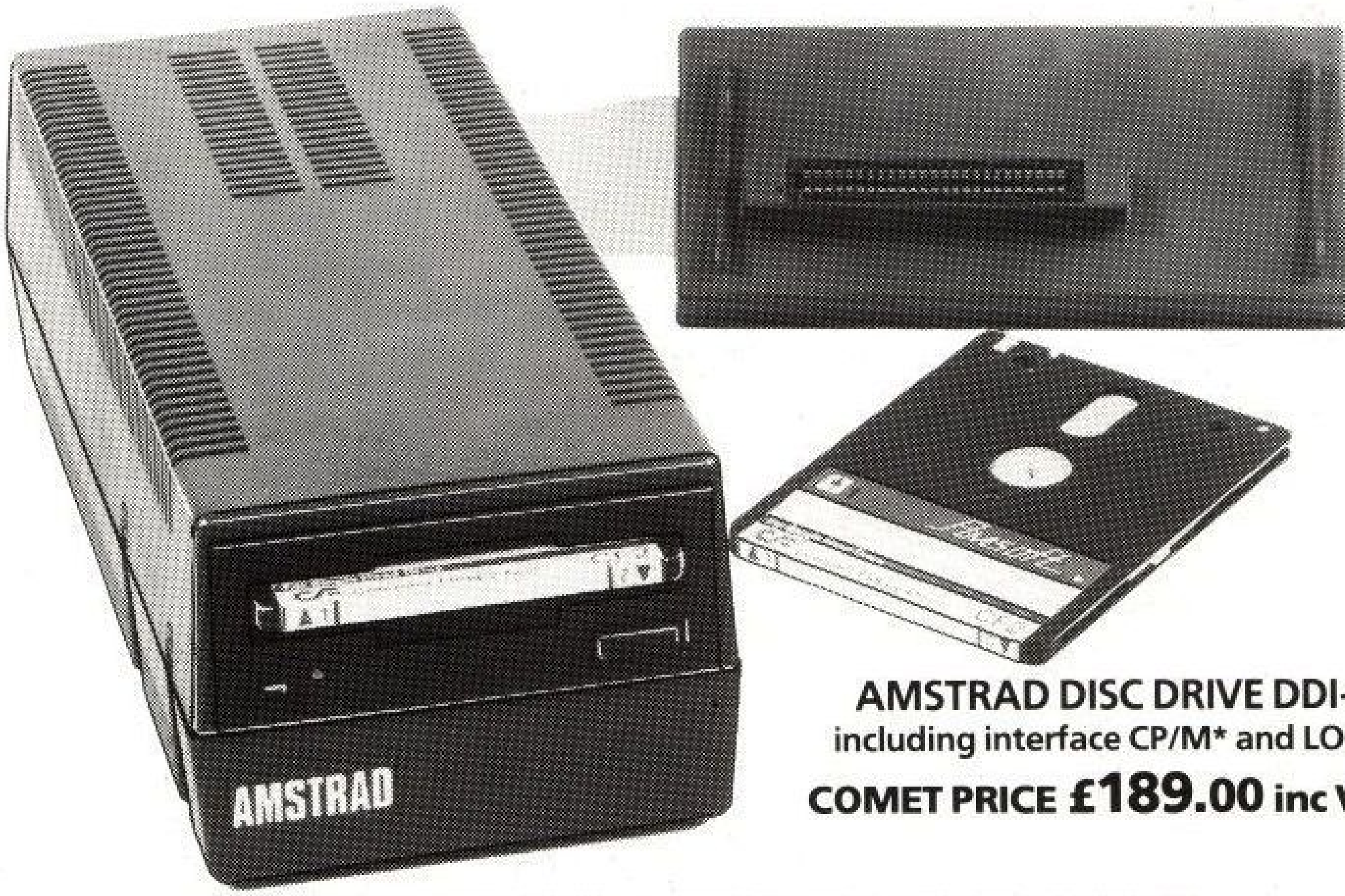
Which means you're not left drumming your fingers while you wait for the information to be loaded.

With the disc you have up to 180k of storage space per side. It retains information more safely too, and takes up a lot less room than the more cumbersome cassette.

Available shortly ... Amstrad FDI disc drive – combines with the DDI-1 to give a dual drive capability which doubles storage capacity and enables quick and easy data duplication – a feature particularly useful for business applications.

So it's really no contest . . . whether you use your CPC 464 for pleasure or to help run your business.

Discover the Amstrad disc drive at Comet. And discover our discounts too!



AMSTRAD DISC DRIVE DDI-1
including interface CP/M* and LOGO
COMET PRICE £189.00 inc VAT

Comet stock a full range of:— Amsoft games software, Amsoft programming aids, Amsoft books, Amsoft business and educational software, Amstrad matched joysticks.

COMET

BRANCHES THROUGHOUT THE U.K.

For the address of your nearest Comet branch see your Local Telephone Directory or ring Teledata 24-hour service on 01-200 0200.

*Trade Mark Digital Research

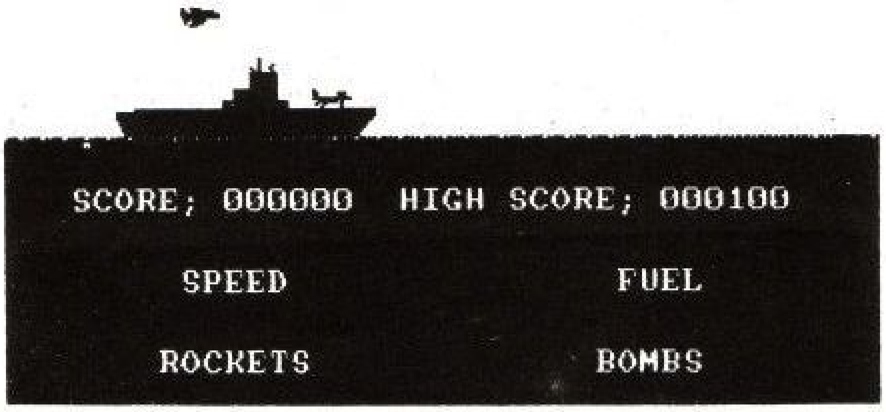
```
10 REM                *-----------------*
20 REM                *  LISTING   2    *
30 REM                *-----------------*
40 MODE 2
50 GOSUB 270
60 INPUT "Load address :",addr
70 lin$=""
80 WHILE lin$<>"END"
90  PRINT HEX$(addr,4);":";:LINE INPUT lin$
100  lin$=UPPER$(lin$)
110  chk=0:oldaddr=addr
120  FOR a=1 TO LEN(lin$) STEP 3
130   m$=MID$(lin$,a,1)
140   l$=MID$(lin$,(a+1),1)
150   m=ASC(m$)
160   l=ASC(l$)
170  IF m>&2F AND m<&3A THEN m=m-&30 ELSE IF m>&40 AND m<&47 THEN m=m-&37 ELSE
 PRINT "error",m$
180  IF l>&2F AND l<&3A THEN l=l-&30 ELSE IF l>&40 AND l<&47 THEN l=l-&37 ELSE
 PRINT "error",l$
190  byt=m*16+l
200  IF a=LEN(lin$)-1 THEN GOTO 240
210  chk=(chk+byt) MOD 256
220  POKE addr,byt
230  addr=addr+1
240  NEXT a
250  IF chk<>byt THEN PRINT "Checksum Error - retype line":a=LEN(lin$):addr=
oldaddr
260 WEND
270 '*------------------------------------------*
280 '*  Instructions - can be omitted, just      *
290 '*  delete 50 and ignore the following.      *
300 '*------------------------------------------*
310 PRINT,"       Machine code loader"
320 PRINT,"       ======= ==== ======"
330 PRINT
340 PRINT"When asked, input address at which to start loading the code"
350 PRINT"this is taken as a decimal value - to use hex, prefix with &."
360 PRINT
370 PRINT"Then enter lines of bytes, each separated by a space, the final"
380 PRINT"one being the checksum for that line."
390 PRINT
400 PRINT"example:-"
410 PRINT
420 PRINT"load address:&8000"
430 PRINT"8000:01 02 03 04 0A      (where 0A is the checksum)"
440 PRINT"8004: etc."
450 PRINT
460 PRINT"If an error occurs, 'Checksum Error - retype line' will be printed"
470 PRINT
480 PRINT "example:-"
490 PRINT
500 PRINT"8000:01 02 03 04 01 01 01 37"
510 PRINT"Checksum Error - retype line"
520 PRINT"8000:01 02 03 04 01 01 01 0D"
530 PRINT"8007: etc."
540 PRINT
550 RETURN
```

DMP1VERSION

```
8ØØØ  21 E1 E9 22 3Ø ØØ F7 EB 1F
8ØØ8  21 3Ø Ø1 19 4E 23 46 79 9B
8Ø1Ø  BØ 28 15 E5 6Ø 69 19 E5 99
8Ø18  4E 23 46 6Ø 69 19 44 4D 2A
8Ø2Ø  E1 71 23 7Ø E1 23 18 E4 E5
8Ø28  21 49 ØØ Ø6 81 ØE ØØ 11 1Ø
8Ø3Ø  2F ØØ CD D7 BC C9 C5 D5 F2
8Ø38  E5 F5 3E Ø9 CD 1E BB 28 EF
8Ø4Ø  ØA 3E 17 CD 1E BB 28 Ø3 3Ø
8Ø48  CD 59 ØØ F1 E1 D1 C1 C9 53
8Ø5Ø  ØØ ØØ ØØ ØØ 7F ØØ ØØ ØØ 7F
8Ø58  ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
8Ø6Ø  DD 21 52 ØØ DD 36 Ø1 ØØ 64
8Ø68  DD 36 Ø2 ØØ DD 36 Ø3 9Ø BB
8Ø7Ø  DD 36 Ø4 Ø1 DD 36 Ø5 ØØ 3Ø
8Ø78  DD 36 ØØ ØØ CD 2E BD 38 Ø3
8Ø8Ø  FB 3E 1B CD 2B BD 3E 4B 92
8Ø88  CD 2B BD 3E Ø2 CD 2B BD AA
8Ø9Ø  3E 4Ø CD 2B BD DD 36 ØØ 46
8Ø98  ØØ DD 36 Ø6 ØØ DD CB ØØ C1
8ØAØ  2E DD 5E Ø1 DD 56 Ø2 DD 7C
8ØA8  6E Ø3 DD 66 Ø4 CD FØ BB 3Ø
8ØBØ  B7 28 Ø4 DD CB ØØ F6 DD 5E
8ØB8  34 Ø6 DD 7E Ø6 FE Ø7 28 C8
8ØCØ  12 DD 6E Ø3 DD 66 Ø4 2B D2
8ØC8  2B DD 75 Ø3 DD 74 Ø4 18 ED
8ØDØ  CC 18 A9 CD 2E BD 38 FB 78
8ØD8  DD 7E ØØ CD 2B BD DD 34 21
8ØEØ  Ø1 DD 34 Ø1 2Ø Ø3 DD 34 47
8ØE8  Ø2 DD 7E Ø1 D6 82 2Ø Ø7 DD
8ØFØ  DD 7E Ø2 FE Ø2 28 12 DD 74
8ØF8  6E Ø3 DD 66 Ø4 11 ØC ØØ D5
81ØØ  19 DD 75 Ø3 DD 74 Ø4 18 DB
81Ø8  8C DD 34 Ø5 3E ØA CD 2B E2
811Ø  BD 3E ØD CD 2B BD 3E 15 1Ø
8118  CD 1E BB 2Ø ØF DD 36 Ø1 E9
812Ø  ØØ DD 36 Ø2 ØØ DD 7E Ø5 75
8128  FE 22 2Ø A5 3E ØF CD 2B 2A
813Ø  BD 3E ØD CD 2B BD C9 22 A8
8138  ØØ 29 ØØ 42 ØØ 5B ØØ ØØ C6
814Ø  ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

EPSONVERSION

```
8ØØØ  21 E1 E9 22 3Ø ØØ F7 EB 1F
8ØØ8  21 3F Ø1 19 4E 23 46 79 AA
8Ø1Ø  BØ 28 15 E5 6Ø 69 19 E5 99
8Ø18  4E 23 46 6Ø 69 19 44 4D 2A
8Ø2Ø  E1 71 23 7Ø E1 23 18 E4 E5
8Ø28  21 49 ØØ Ø6 81 ØE ØØ 11 1Ø
8Ø3Ø  2F ØØ CD D7 BC C9 C5 D5 F2
8Ø38  E5 F5 3E Ø9 CD 1E BB 28 EF
8Ø4Ø  ØA 3E 17 CD 1E BB 28 Ø3 3Ø
8Ø48  CD 59 ØØ F1 E1 D1 C1 C9 53
8Ø5Ø  ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
8Ø58  ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
8Ø6Ø  DD 21 52 ØØ DD 36 Ø1 ØØ 64
8Ø68  DD 36 Ø2 ØØ DD 36 Ø3 9Ø BB
8Ø7Ø  DD 36 Ø4 Ø1 DD 36 Ø5 ØØ 3Ø
8Ø78  DD 36 ØØ ØØ 3E 1B CD 2B 64
8Ø8Ø  BD 3E 41 CD 2B BD 3E Ø6 35
8Ø88  CD 2B BD CD 2E BD 38 FB AØ
8Ø9Ø  3E 1B CD 2B BD 3E 4B CD 64
8Ø98  2B BD 3E 4Ø CD 2B BD 3E 59
8ØAØ  Ø1 CD 2B BD DD 36 ØØ ØØ C9
8ØA8  DD 36 Ø6 ØØ DD CB ØØ 26 E7
8ØBØ  DD 5E Ø1 DD 56 Ø2 DD 6E BC
8ØB8  Ø3 DD 66 Ø4 CD FØ BB B7 79
8ØCØ  28 Ø4 DD CB ØØ C6 DD 34 AB
8ØC8  Ø6 DD 7E Ø6 FE Ø7 28 12 A6
8ØDØ  DD 6E Ø3 DD 66 Ø4 2B 2B EB
8ØD8  DD 75 Ø3 DD 74 Ø4 18 CC 8E
8ØEØ  18 A9 CD 2E BD 38 FB DD 89
8ØE8  7E ØØ CD 2B BD DD 34 Ø1 45
8ØFØ  DD 34 Ø1 2Ø Ø3 DD 34 Ø2 48
8ØF8  DD 7E Ø1 D6 82 2Ø Ø7 DD B8
81ØØ  7E Ø2 FE Ø2 28 12 DD 6E Ø5
81Ø8  Ø3 DD 66 Ø4 11 ØC ØØ 19 8Ø
811Ø  DD 75 Ø3 DD 74 Ø4 18 8C 4E
8118  DD 34 Ø5 3E ØA CD 2B BD 13
812Ø  3E ØD CD 2B BD 3E 15 CD 2Ø
8128  1E BB 2Ø ØF DD 36 Ø1 ØØ 1C
813Ø  DD 36 Ø2 ØØ DD 7E Ø5 FE 73
8138  22 2Ø A5 3E ØF CD 2B BD E9
814Ø  3E ØD CD 2B BD C9 22 ØØ EB
8148  29 ØØ 42 ØØ 5B ØØ ØØ ØØ C6
815Ø  ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```



"ELECTRIC EDDY"

"Electric Eddy" is a modern day version of Hangman. It has an on board memory of two hundred and forty words, which range from nail to paraphernalia. As well as these standard features, it has the facility to enter your own word list, or even challenge another player with a word of your own choice. Whichever of these features you choose, you then have the option of a time limit.

By now, you may be wondering why the game is called "Electric Eddy" instead of "Hangman" or some similar name. The secret lies in the man. He isn't hung. I'm sure you can figure out what does happen to him.

```
10 ' ************************************
   ***********************************
20 ' *********** Programs from AMSTRAD C
   PC464 User Magazine *************
30 ' ********************** ELECTRIC ED
   DY *******************************
40 ' ************************************
   ***********************************
50 MODE 2
60 MODE 0:DIM fail$(26),Gword$(26),word$
   (15),gues$(15),d$(15),mem$(260),v$(4)
   :INK 0,0:BORDER 13:v=1:INK 1,13:INK 2
   ,16:INK 3,0:INK 4,1:INK 5,15:INK 6,3:
   INK 7,1:INK 8,12:PEN #1,6:PEN #2,7:PE
   N #3,8
70 FOR z=1 TO 20
80 PLOT 0,z,1:DRAW z,0:PLOT 0,400-z:DRAW
    z,400:PLOT 640-z,0:DRAW 640,z:PLOT 6
   40-z,400:DRAW 640,400-z
90 NEXT z:WINDOW #4,1,14,3,23:WINDOW #5,
   1,20,4,24
100 PRINT #1:PRINT #1,"     INSTRUCTIONS"
    :LOCATE 1,5:PRINT "  AFTER COMPLETIN
    G  ":PRINT "THE INSTRUCTIONS THE ":P
    RINT "  MENU WILL APPEAR.  ":PRINT "
       TO SELECT YOUR       ":PRINT "  OPT
    ION PRESS THE  ":PRINT "   INDICATED
      KEY ":LOCATE #2,1,17
110 PRINT #2," THE NUMBER BEFORE!  ":PRI
    NT #2,"      THE OPTION":LOCATE 1,22:
    PRINT "  PRESS   SPACE BAR  ":PRINT "
       TO  CONTINUE"
120 IF INKEY$<>" " THEN 120
130 CLS #5:LOCATE 1,5:PRINT "  IF YOU CH
    OOSE AN  ":PRINT " OPTION BETWEEN 1-
    4  ":PRINT "YOU WILL GO STRAIGHT ":P
    RINT "    TO":LOCATE #2,8,11:PRINT #2
    ,"TIME LIMIT":PRINT:PRINT "   YOU NOW
      HAVE TO    ":PRINT " PRESS Y OR N W
    HICH  ":PRINT "STAND FOR  YES OR NO
140 LOCATE 1,22:PRINT "   PRESS   SPACE BA
    R  ":PRINT "      TO CONTINUE"
150 IF INKEY$<>" " THEN 150
160 CLS #5:LOCATE 1,5:PRINT " IF YOU PRE
    SS Y YOU ":PRINT "   WILL BE   ASKED
    ":LOCATE #2,7,9:PRINT #2,"HOW LONG":
    LOCATE 1,11:PRINT "  YOU TYPE IN YOU
    R  ":PRINT "  REPLY IN SECONDS. ":PR
    INT " THEN THE GAME WILL ":PRINT "
         START."
170 LOCATE 1,22:PRINT "   PRESS   SPACE BA
    R  ":PRINT "      TO CONTINUE"
180 IF INKEY$<>" " THEN 180
190 CLS #5:LOCATE 1,5:PRINT "HOWEVER IF
    YOU CHOSE":PRINT "    OPTIONS 5 OR 6
      ":PRINT "  BEFORE YOU GO TO   ":LOC
    ATE #2,5,11:PRINT #2,"TIME LIMIT":LO
    CATE 1,13:PRINT "  YOU HAVE   OPTION
     ":PRINT "     INSTRUCTIONS     ":PRIN
    T " YOU WILL  FIND OUT  "
200 PRINT "ABOUT THESE WHEN YOU":PRINT "
         SELECT  THEM     ":LOCATE #1,1,24
    :PRINT #1,"PRESS SPACE TO START"
210 IF INKEY$<>" " THEN 210 ELSE CLS #5:
    LOCATE 1,2:PRINT "                  ":
    GOSUB 1110
220 LOCATE #2,4,2:PRINT #2,"ELECTRIC  ED
    DY"
230 FOR z=1 TO 240:READ d$:mem$(z)=d$:NE
    XT z
240 INK 2,16:INK 4,1:INK 5,15:INK 6,3:GO
    TO 310
250 INK 2,16:INK 4,1:INK 5,15:INK 6,3:go
    =0:c=0
260 CLS #4
270 LOCATE 5,10:PRINT "SAME  AS":PRINT:P
    RINT  "      BEFORE":in$=INKEY$
280 IF in$="n" OR in$="N" THEN q=0:CLS #
    4:GOTO 310
290 IF in$="y" OR in$="Y" THEN q=1:GOTO
    330
300 GOTO 270
310 LOCATE 1,7:PRINT " 1 LIST A":PRINT:P
    RINT " 2 LIST B":PRINT:PRINT " 3 LIS
    T C":PRINT:PRINT " 4 LIST D":PRINT:P
    RINT " 5 OWN LIST":PRINT:PRINT " 6 C
    HALLENGE"
320 v$=INKEY$:IF v$="" THEN 320
330 IF v$<>"" THEN CLS #4
340 FOR z=1 TO 170
350 IF z<27 THEN fail$(z)=""
360 IF z<27 THEN Gword$(z)=""
```

ZAXXON FORT APOCALYPSE BEACH HEAD
POLE POSITION CONGO BONGO
SPITFIRE ACE SOLO FLIGHT
UP 'N DOWN SPY HUNTER
BRUCE LEE DROP ZONE
BLUE MAX RAID OVER
STELLA 7 MOSCOW
PAC MAN F15 STRIKE
SENTINEL EAGLE MR.
FLAK NATO DO DRELBS
COMMANDER KNIGHTS OF
BUCK ROGERS THE DESERT
INDIANA JONES CONAN SLINKY
TAPPER FORBIDDEN FOREST DALLAS QUEST

U.S. GOLD
All American Software

GOLD STRIKE

GUARANTEED PAYMENTS FOR FIRST CLASS ASSEMBLY LANGUAGE PROGRAMMERS

Translation work on the chart-topping U.S. Gold Commodore games to the

Amstrad CPC 64.

★★★★★★★★★★

Become part of the best motivated team in the U.K. . . . one-off or long term contracts available and original game or software concepts also required for World Wide publication.

★★★★★★★★★★

For fame and fortune from U.S. Gold
write in confidence to:
Software Development, U.S. Gold,
6 Central Street, Manchester M2 5NS,
or phone 061-832 6633 for details.

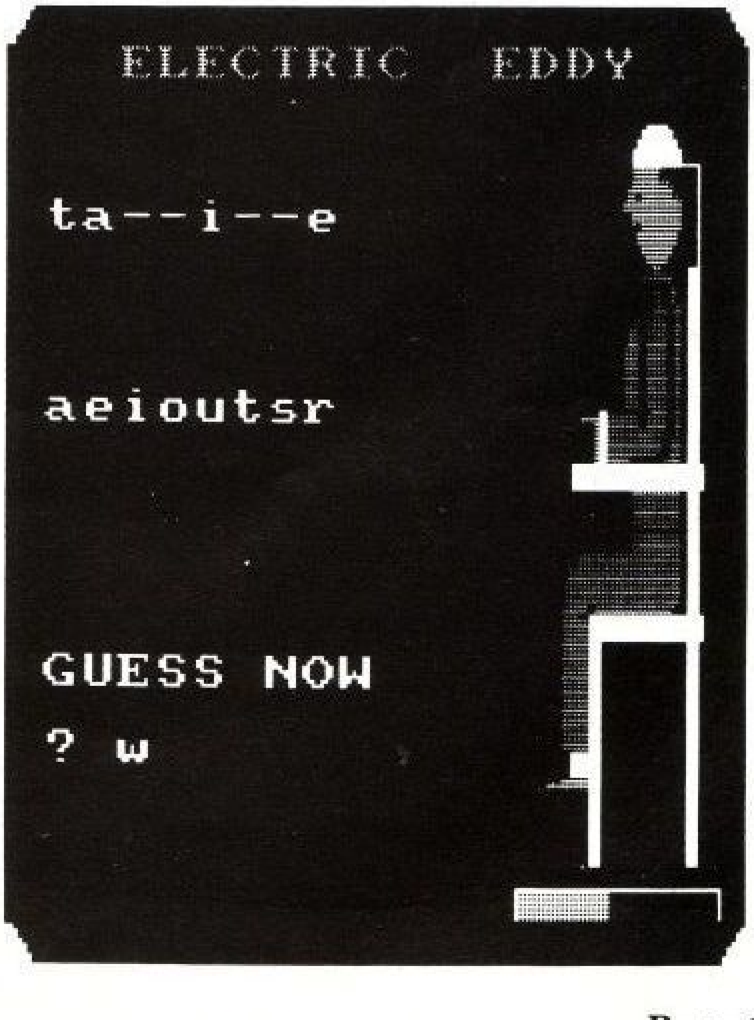
```
370 PLOT 615-z,16,3:DRAW 615-z,28:NEXT z
380 LOCATE 2,10:PRINT "                ":I
    F v$="5" AND q=1 OR v$="5" AND gues$
    ="new word" THEN 530
390 IF v$="1" THEN 470
400 IF v$="2" THEN 480
410 IF v$="3" THEN 490
420 IF v$="4" THEN 500
430 IF v$="5" THEN 510
440 IF v$="6" THEN 570
450 GOTO 310
460 FOR p=1 TO 1500:NEXT p:GOTO 310
470 r=RND*60:word$=mem$(r+1):GOTO 600
480 r=RND*60:word$=mem$(r+60):GOTO 600
490 r=RND*60:word$=mem$(r+120):GOTO 600
500 r=RND*60:word$=mem$(r+180):GOTO 600
510 LOCATE 2,4:PRINT "number of":PRINT "
      words":LOCATE 2,7:INPUT l:CLS #4
520 FOR z=1 TO l:LOCATE 2,10:INPUT a$(z)
    :LOCATE 2,10:CLS #4:NEXT z
530 r=INT(RND*l)
540 IF r=0 THEN 530
550 word$=a$(r)
560 IF word$="" THEN 530 ELSE 600
570 LOCATE 2,6:PRINT "ENTER WORD":LOCATE
     2,8:INPUT word$:word$=LOWER$(word$)
580 IF word$="" THEN LOCATE 2,8:PRINT "
              ":GOTO 570
590 LOCATE 2,6:PRINT "           ":PRINT:
    PRINT "           "
600 LOCATE 3,10:PRINT "TIME LIMIT":in$=I
    NKEY$
610 IF in$="n" OR in$="N" THEN 640
620 IF in$="y" OR in$="Y" THEN 630 ELSE
    600
630 LOCATE 2,12:PRINT "           ":LOCAT
    E 3,10:PRINT "          ":LOCATE 4,1
    0:PRINT "HOW LONG":LOCATE 2,12:INPUT
     t:IF t>600 THEN 630 ELSE LOCATE 2,1
    2:PRINT "         ":t=t*50:AFTER t,1 G
    OSUB 1710
640 LOCATE 3,10:PRINT "           "
650 length=LEN(word$):go=0:c=0
660 FOR a=1 TO length:LOCATE 1+a,6:PRINT
     "-";:NEXT a
670 LOCATE 2,18:PRINT "GUESS NOW":LOCATE
     2,20
680 INPUT gues$:g=1
690 gues$=LOWER$(gues$)
700 LOCATE 2,18:PRINT "           ":PRINT:
    PRINT "           "
710 IF gues$="" THEN 670
720 IF gues$=word$ THEN 1010
730 IF LEN(gues$)=1 THEN 750
740 SOUND 1,300,30,5,10:LOCATE #1,4,18:P
    RINT #1,"WRONG":FOR p=0 TO 500:NEXT
    p:g=1:GOTO 830
750 FOR d=1 TO 26:IF gues$=fail$(d) THEN
     800 ELSE NEXT d
760 FOR d=1 TO length
770 a$=MID$(word$,d,1)
780 IF gues$=a$ THEN 930
790 NEXT d:GOTO 830
800 SOUND 1,300,30,5,10,15:LOCATE 2,18:P
    RINT "YOU GUESSED":PRINT "  THAT ONC
    E"
810 FOR p=0 TO 500:NEXT p
820 LOCATE 2,18:PRINT "           ":PRIN
    T "           ":GOTO 670
830 c=c+1
840 IF g=1 THEN go=go+1:ON go GOSUB 1680
    ,1680,1680,1680,1680,1680,1680,1680,
    1680,1710
850 IF LEN(gues$)>1 THEN 670
860 fail$(c)=gues$
870 LOCATE 2,11
880 FOR d=1 TO c
890 IF d=10 THEN LOCATE 2,13
900 IF d=18 THEN LOCATE 2,15
910 PRINT fail$(d);
920 NEXT d:GOTO 670
930 g=0:LOCATE d+1,6:PRINT a$:SOUND 1,20
    0,20,5
940 Gword$(d)=a$
950 FOR b=1 TO length
960 IF Gword$(b)=MID$(word$,b,1) THEN NE
    XT b ELSE GOTO 790
970 AFTER 1,1 GOSUB 980
980 ENT 1,2,1,3,4,-1,4,4,1,5,1,-1,6,2,1,
    7:ENV 1,3,60,1,1,0,10,120,-1,8:SOUND
     1,0,10:SOUND 1,200,25,6,1,1:SOUND 1
    ,180,25,6,1,1:SOUND 1,160,25,6,1,1:S
    OUND 1,148,50,6,1,1:SOUND 1,200,55,6
    ,1,1:SOUND 1,148,25,6,1,1:SOUND 1,16
    0,25,6,1,1:SOUND 1,148,25,6,1,1
990 LOCATE #3,1,22:PRINT #3,"CONGRATULAT
    IONS":SOUND 1,135,50,6,1,1:SOUND 1,1
    80,55,6,1,1:SOUND 1,180,25,6,1,1:SOU
    ND 1,160,25,6,1,1:SOUND 1,148,25,6,1
    ,1:SOUND 1,120,30,6,1,1:SOUND 1,135,
    15,6,1,1:SOUND 1,135,25,6,1,1:SOUND
    1,148,25,6,1,1
1000 PRINT "                ":SOUND 1,1
     48,25,6,1,1:SOUND 1,160,25,6,1,1:SO
     UND 1,180,25,6,1,1:SOUND 1,160,25,6
     ,1,1:SOUND 1,148,60,6,1,1:LOCATE 1,
     22:PRINT "                ":GOTO 250
1010 AFTER 1,1 GOSUB 1020
1020 INK 1,13:INK 2,16:INK 3,0:INK 4,1:I
     NK 5,15:INK 6,3:go=0:c=0:LOCATE 2,6
     :PRINT word$
1030 r=INT(RND*3)
1040 IF r=1 THEN LOCATE #3,2,18:PRINT #3
     ,"PRETTY GOOD":GOTO 1070
1050 IF r=2 THEN LOCATE #3,4,18:PRINT #3
     ,"NOT BAD":GOTO 1070
1060 LOCATE #3,2,18:PRINT #3,"OK SHOW OF
     F"
```



GET SORCERY OR YOU'LL TURN INTO A FROG
Virgin
"Get Sorcery or you'll turn into a frog."
Well, that was the verdict of Which Micro on the Commodore 64 version. Now available on the 14th February is a greatly enhanced, stunning new version of Sorcery for the Amstrad and MSX. 40 screens of amazing and breathtaking graphics (50 in the case of the MSX version), with endless hours of tough and challenging game play.
What the press said about the '64' version:
"Get Sorcery or You'll turn into a frog" – Which Micro (CBM 64).
"Virgin's best game since the screen-scorching Falcon Patrol I & II" – Commodore User (CBM 64).
"I have played this game every night since I received it" – TV Gamer (CBM 64).
"Fast, wacky and very polished" – Commodore Horizons (CBM 64).
The World has fallen into the Dark Ages. The Evil Forces of the Necromancer are creeping to every last corner of the ancient land that was once the powerful domain of the Great Sorcerers. You are the last free Sorcerer, all the others have, one by one, been entrapped by the hideous evil that abounds.
So don't delay, get your copy now. It's available now from leading computer stores.
Available for
AMSTRAD CPC 464 £8.95
MSX £8.95
CBM 64 £8.95
SPECTRUM 48K £6.95
ALL SCREEN SHOTS FROM AMSTRAD CPC 464 VERSION.
VGD 9001. JOYSTICK CONTROL
SORCERY
FOR THE AMSTRAD CPC 464
FLASHLOAD

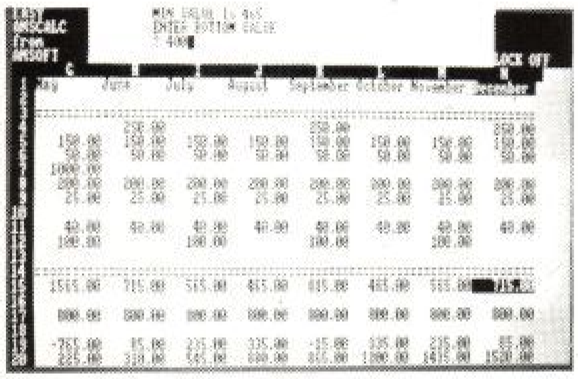
```
1070 ENV 1,3,40,2,1,0,10,120,-1,6:ENV 2,
     3,40,1,1,0,10,120,-1,8:ENT 1,2,1,1,
     4,-1,2,4,1,3,4,-1,4,4,1,5:SOUND 1,2
     00,30,4,1,1:SOUND 1,200,15,4,1,1:SO
     UND 1,200,30,4,1,1:SOUND 1,200,30,4
     ,1,1:SOUND 240,30,4,1,1:SOUND 1,200
     ,35,4,1,1:SOUND 1,150,30,5,1,1
1080 SOUND 1,150,15,5,1,1:SOUND 1,150,30
     ,5,1,1:SOUND 1,150,30,5,1,1:SOUND 1
     ,200,30,5,1,1:SOUND 1,150,35,5,1,1:
     SOUND 1,120,30,6,1,1:SOUND 1,120,15
     ,6,1,1:SOUND 1,120,30,6,1,1:SOUND 1
     ,120,30,6,1,1:SOUND 1,150,30,6,1,1:
     SOUND 1,120,35,6,1,1
1090 SOUND 1,100,120,6,2,1
1100 LOCATE 2,18:PRINT "                ":G
     OTO 250
1110 FOR z=0 TO 10
1120 IF z<8 THEN PLOT 597-z,40,1:DRAW 59
     7-z,290
1130 IF z<8 THEN PLOT 512-z,40:DRAW 512-
     z,135
1140 PLOT 505,135+z:DRAW 600,135+z
1150 NEXT z
1160 FOR z=1 TO 360 STEP 4
1170 DEG
1180 ORIGIN 560,320
1190 DRAW 22*COS(z),30*SIN(z),2
1200 IF z<150 THEN DRAW 20*COS(z),10*SIN
     (z),5
1210 NEXT z
1220 ORIGIN 0,0:PLOT 538,325,2:DRAW 534,
     315:PLOT 538,315:DRAW 538,320:PLOT
     545,320,3
1230 FOR z=0 TO 270 STEP 3
1240 ORIGIN 560,270
1250 IF z<180 THEN DRAW 20*COS(z),20*SIN
     (z),4
1260 ORIGIN 573,156
1270 IF z>180 THEN  DRAW 10*COS(z+90),10
     *SIN(z+90)
1280 ORIGIN 0,0
1290 IF z<10 THEN PLOT 581-z,270:DRAW 58
     1-z,156
1300 IF z>20 AND z<50 THEN PLOT 520+z,27
     0:DRAW 520+z,146
1310 ORIGIN 536,236
1320 IF z>100 AND z<260 THEN DRAW 5*COS(
     z),30*SIN(z)
1330 NEXT z
1340 FOR z=0 TO 270 STEP 3
1350 ORIGIN 560,275
1360 IF z<180 THEN PLOT 11*COS(z),11*SIN
     (z),3
1370 ORIGIN 560,220
1380 IF z>180 THEN PLOT 11*COS(z+90),11*
     SIN(z+90)
1390 NEXT z
1400 ORIGIN 0,0:PLOT 572,275:DRAW 572,22
     0:PLOT 550,275:DRAW 550,224:PLOT 53
     2,230:DRAW 553,230:PLOT 557,210:DRA
     W 536,210
1410 FOR z=0 TO 18
1420 PLOT 537,211+z:DRAW 520,211+z,4
1430 IF z>8 THEN PLOT 490,190+z,1:DRAW 6
     00,190+z
1440 IF z>12 THEN PLOT 500+z,210:DRAW 50
     0+z,230
1450 NEXT z
1460 PLOT 508,211,2:DRAW 508,229:DRAW 49
     8,229:PLOT 507,227:DRAW 498,227
1470 FOR z=90 TO 180
1480 ORIGIN 0,0
1490 IF z<114 THEN PLOT 540,260-z:DRAW 5
     04,260-z,4
1500 ORIGIN 500,148
1510 DRAW 22*COS(z),22*SIN(z),4
1520 NEXT z
1530 ORIGIN 0,0:PLOT 540,180:DRAW 580,18
     0,3
1540 FOR z=1 TO 60
1550 PLOT 480,148-z:DRAW 500,148-z,4
1560 NEXT z
1570 PLOT 500,144,3:PLOT 497,144:PLOT 50
     4,146:PLOT 504,148
1580 FOR z=1 TO 10
1590 PLOT 490,88-z,1:DRAW 500,88-z
1600 NEXT z
1610 PLOT 500,77,2:DRAW 480,77:PLOT 470,
     74:DRAW 500,74:PLOT 500,73:DRAW 460
     ,73
1620  PLOT 440,18:DRAW 440,30,1:DRAW 619
     ,30:DRAW 619,18
1630 FOR z=1 TO 180 STEP 3
1640 ORIGIN 560,335
1650 DRAW 21*COS(Z),17*SIN(Z),1
1660 NEXT Z
1670 ORIGIN 0,0:PLOT 575,335:DRAW 598,33
     5:DRAW 598,290:RETURN
1680 SOUND 1,1500,30,7:FOR z=1 TO 19
1690 PLOT go*19+425+z,28,6:DRAW go*19+42
     5+z,18
1700 NEXT z:RETURN
1710 AFTER 1,1 GOSUB 1720
1720 FOR z=445 TO 615:PLOT z,18,6:DRAW z
     ,28:NEXT z
1730 INK 2,16,13:INK 4,26,1:SPEED INK 3,
     2
1740 ENV 1,2,60,1,1,0,10,120,-1,6:ENV 2,
     2,60,1,1,0,60,120,-1,6:ENT 1,6,1,2,
     12,-1,2,12,1,2,6,-1,2:SOUND 1,1000,
     80,6,1,1:SOUND 1,1000,70,6,1,1:SOUN
     D 1,1000,25,6,1,1:SOUND 1,1000,70,6
     ,1,1:SOUND 1,900,70,6,1,1:SOUND 1,9
     50,30,6,1,1:SOUND 1,950,50,6,1,1
1750 SOUND 1,1000,30,6,1,1:SOUND 1,1000,
     50,6,1,1:SOUND 1,1050,30,6,1,1:SOUN
     D 1,1000,140,6,2,1:SOUND 1,900,80,6
```

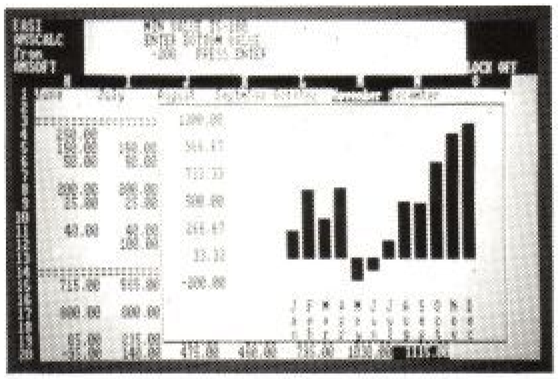
```
     ,1,1:SOUND 1,900,70,6,1,1:SOUND 1,9
     00,25,6,1,1:SOUND 1,900,70,6,1,1:SO
     UND 1,700,70,6,1,1:SOUND 1,800,30,6
     ,1,1:SOUND 1,800,50,6,1,1
1760 INK 2,0:INK 4,0:INK 5,0:LOCATE #1,2
     ,6:PRINT #1,word$
1770 SOUND 1,900,30,6,1,1:SOUND 1,900,50
     ,6,1,1:SOUND 1,950,30,6,1,1:SOUND 1
     ,900,140,6,2,1:SOUND 1,530,50,6,1,1
     :SOUND 1,600,30,6,1,1:SOUND 1,680,5
     0,6,1,1:SOUND 1,700,30,6,1,1:SOUND
     1,700,70,6,1,1:SOUND 1,900,70,6,1,1
     :SOUND 1,530,50,6,1,1
1780 LOCATE 2,18:PRINT "              ":LOC
     ATE 2,20:PRINT "              ":LOCA
     TE #1,2,18:PRINT #1,"YOU LOST":PRIN
     T "         "
1790 SOUND 1,600,30,6,1,1:SOUND 1,680,50
     ,6,1,1:SOUND 1,700,30,6,1,1:SOUND 1
     ,700,70,6,1,1:SOUND 1,900,70,6,1,1:
     SOUND 1,1000,80,6,1,1:SOUND 1,1000,
     70,6,1,1:SOUND 1,1000,25,6,1,1:SOUN
     D 1,1000,70,6,1,1:SOUND 1,900,70,6,
     1,1:SOUND 1,950,30,6,1,1
1800 SOUND 1,950,50,6,1,1:SOUND 1,1000,3
     0,6,1,1:SOUND 1,1000,50,6,1,1:SOUND
      1,1050,30,6,1,1:SOUND 1,1000,140,6
     ,2,1:LOCATE 2,18:PRINT "          ":G
     OTO 250
1810 DATA apple,arch,baby,balloon,carry,
     chopper,darling,dazzle,earth,employ
     ,father,feast,gangster,glimmer,happ
     y,hectic,ideal,important,jacket,jun
     gle,keeper,kettle,large,leader,mach
     ine,market,nail,nerve,orange,observ
     e,paper,pardon,quack,quiet,race,rap
     id
1820 DATA sacred,scamper,tablet,teacher,
     ugly,understand,vacant,verse,wade,w
     all,yell,yard,zip,zero,aim,area,bea
     ch,behave,catch,cattle,desert,detai
     l,elderly,employ,arrival,apology,ba
     ckwards,bankrupt,calendar,challenge
     ,dangerous,dedication,effortless,el
     astic
1830 DATA familiar,fathom,gaiety,garrulo
     us,habitat,harbour,idiot,impudent,j
     ettison,juvenile,knight,keepsake,la
     bel,laughable,magician,management,n
     aughty,neighbour,obey,obnoxious,pai
     nful,parchment,qualify,quarrel,radi
     ant,ration,salary,savior,tactful
1840 DATA telepathy,umpire,uproad,vacati
     on,vegetable,walk,wander,year,yeste
     rday,zest,zone,alibi,arena,business
     ,bursar,cause,cleanse,olepth,desola
     tion,electrify,embroider,autumn,arc
     haic,bamboozle,blasphemy,campaign,c
     haperon,dastardly,debonair,ecstatic
1850 DATA endeavour,facetious,fatuous,ga
     lvanize,gauge,hallucination,height,
     ignite,impromptu,jewellery,juncture
     ,kitchen,kerchief,lacerate,language
     ,maintenance,malnutrition,negotiate
     ,nonchalant,oblivion,obsession,paci
     fy,pariah,quarter,quirk,raconteur
1860 DATA rapacious,sabotage,scrpulous,t
     alkative,technique,unbiased,undisti
     nguished,velocity,vault,warranty,wa
     rrior,yield,yearning,zealows,zenith
     ,agriculture,avenue,buoyant,burlesq
     ue,catastrophe,characteristic,desce
     ndant,dexterity,eligible,embarrassi
     ng
1870 DATA austere,annul,banal,bazaar,cam
     ouflage,chauvinism,daub,deceit,effi
     gy,encyclopaedia,feasible,feign,gha
     stly,gnarled,hirsute,hysteria,illeg
     ible,inertia,jeopardy,jaundiced,kni
     ck-knack,kudos,laconic,lascivious,m
     agenta,marauder,nebulous,nymph,obli
     qe
1880 DATA opuleuce,pamphlet,paraphernali
     a,queasy,quixotic,raucous,rhythm,sc
     intillating,sepulchre,tangible,tawd
     ry,unique,untopia,vague,vehicle,wei
     gh,whimsica,yokel,yowl,zephyr,zealo
     t,aisle,avalanche,beleaguer,bureau,
     cemetery,coherent,diabolical,dialog
     ue
```

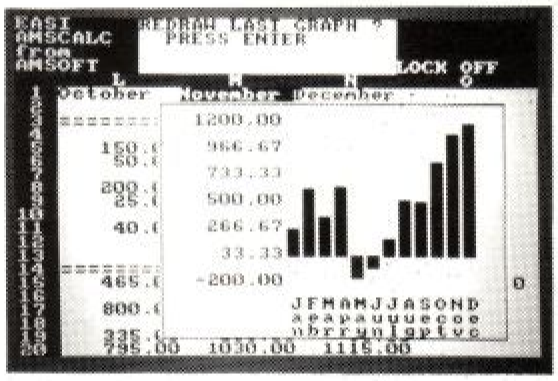


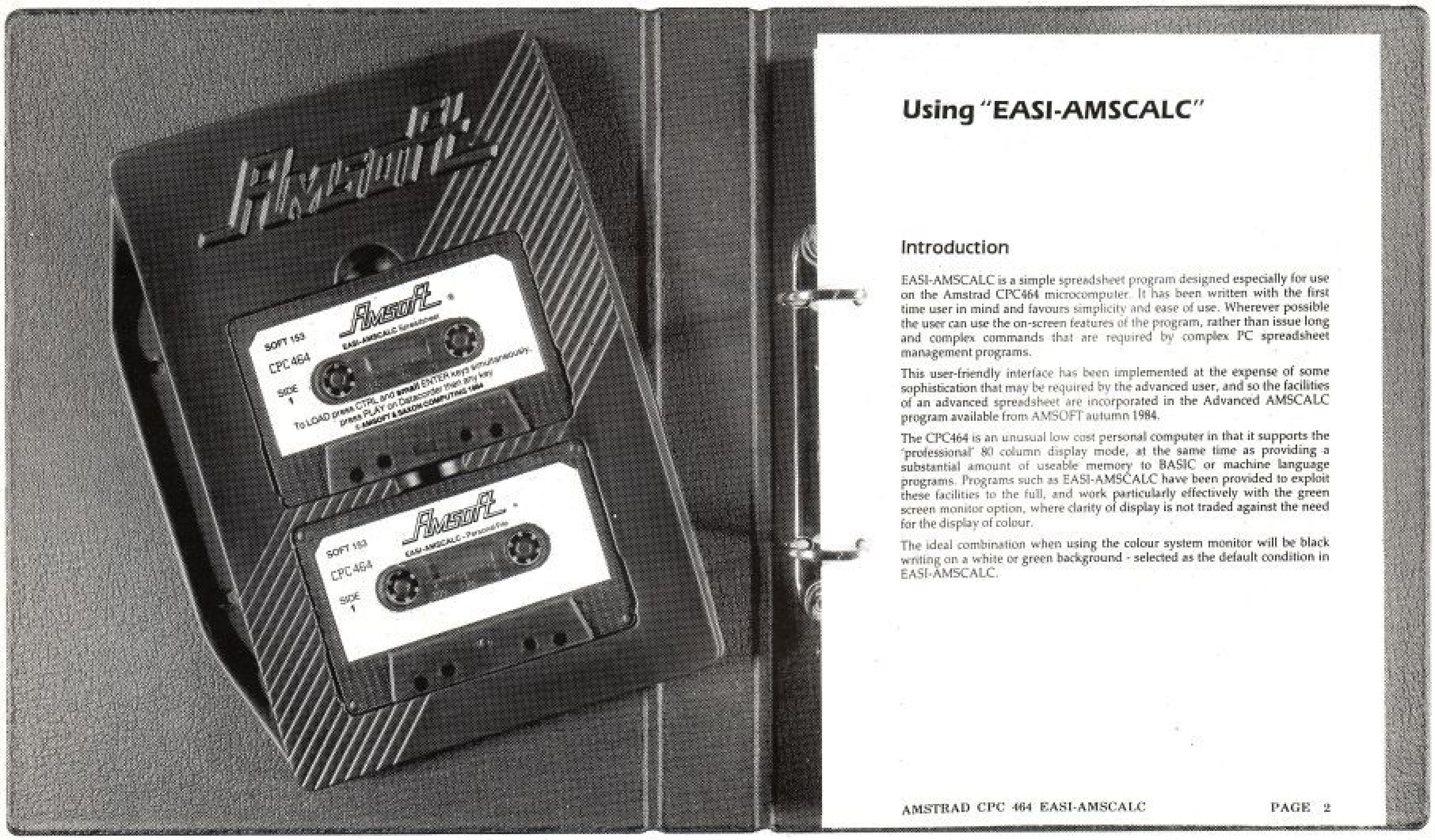
Easi-Amscalc

A low cost computer spreadsheet that includes a linked bar chart graphics option. Very simple to use and understand, yet capable of producing accounts, balance sheets, cash flow analysis, project costings etc.









Easi-Amscalc includes features to save and retrieve files from tape, list to a printer, jump directly to an entry anywhere on the sheet, replicate, rowsum, columnsum etc.

Easi-Amscalc as been specifically written for the CPC464 to make the best use of the hardware facilities. Selectable colour combinations, 40 or 80 column display.

Supplied complete with ring binder, library slip case and full instructions including worked examples.

SOFT 153 Easi-Amscalc computer spreadsheet £19.95

Manipulating the Amstrad CPC464 Screen Display

by David Looker, DJL Software

The Amstrad CPC464 has been widely acclaimed as the most exciting machine to enter the competitive home computer market during 1984, and I would be one of the first to agree with this statement.

However, in order to explore the possibilities offered by the high resolution colour graphics modes of the CPC464 to the full, it is necessary to investigate the way in which the screen display is organised.

The purpose of this article is to explain, in detail, the way in which the screen can be accessed and manipulated in order to produce the exciting visual effects of which this machine is capable.

Screen Display Modes

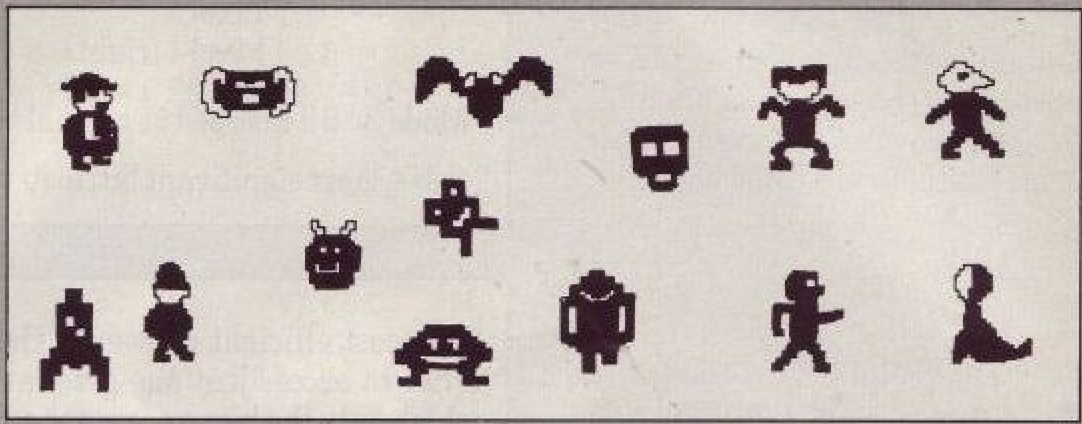
The CPC464 is capable of three display modes, as follows:

MODE 0: Text 20 columns x 25 rows, Graphics 160 x 200 pixels, individually addressable in 16 colours out of 27.

MODE 1: Text 40 columns x 25 rows, Graphics 320 x 200 pixels, individually addressable in 4 colours out of 27.

MODE 2: Text 80 columns x 25 rows, Graphics 640 x 200 pixels, individually addressable in 2 colours out of 27.

Text and Graphics can be mixed on the screen in any mode, but it is not normally possible to have more than one screen mode in operation at the same time. (This can be achieved by accessing the display controller directly, using interrupt-based techniques, but the programming required is fairly



complex, and is beyond the scope of this article.)

Each character 'cell' is 8 pixels wide by 8 pixel-lines high; the height of a character remains constant regardless of screen mode, whereas the width of a character varies according to the horizontal resolution of the currently selected mode.

Screen Memory Mapping

The screen display occupies 16K bytes in the top of the memory map, from C000H (49152 decimal) to FFFFH (65535 decimal).

This area is effectively split up into 8 blocks of 2K bytes, with the first 2K block containing the top pixel lines of each of the 25 character rows, the second block containing the 2nd pixel lines of all 25 character rows, and so on.

Each pixel line is 80 bytes in length, regardless of screen mode; since 80 x 25=2000, and 2K=2048 bytes, there are 48 unused bytes at the end of each 2K block.

From the aspect of screen addressing, however, the odd 48 bytes are not important. In order to calculate the address of the byte immediately below any byte in the screen, simply add 0800H (2048 decimal) to the first address, provided that the result is not greater than FFFFH (65535 decimal). If it is, then subtract 3FB0H (16304 dec) from the total.

A typical example of this technique is demonstrated in the following routine, which is designed to store an area of the screen 8 bytes wide by 8 pixel lines high, into a 64 byte block at some convenient location in memory:

```
START    LD   HL, XXXX      ; Address of top left corner of area
         LD   DE, XXXX      ; Address of store area
         LD   B, 8          ; Number of vertical pixel lines
LOOP1    PUSH BC
         PUSH HL
         LD   B, 8          ; Number of horizontal bytes
LOOP2    LD   A, (HL)
         LD   (DE), A
         INC  DE
         INC  HL
         DJNZ LOOP 2
         POP  HL
         LD   BC, #0000     ; Add 0800H to screen address
         ADD  HL, BC        ;
         JR   NC, LOOP END  ; Jump if total not more than FFFFH
         LD   BC, 3FB0H     ;
         AND  A             ; Else subtract 3FB0H
         SBC  HL, BC        ;
LOOPEND  POP  BC
         DJNZ LOOP 1
         RET
```

Pixel Coding

The number of pixels in a byte will vary according to the screen mode: 2 pixels per byte in Mode 0, 4 per byte in Mode 1 and 8 per byte in Mode 2.

The CPC464 Firmware Manual states that 'pixels are encoded in a non-obvious manner'. This is somewhat of an understatement, as is demonstrated below:

Mode 0:	Pixel 1 (left)	Bits 1, 5, 3, 7 (msb to lsb)
	Pixel 2 (right)	Bits 0, 4, 2, 6 (msb to lsb)
Mode 1:	Pixel 1 (left)	Bits 3, 7 (msb, lsb)
	Pixel 2	Bits 2, 6 (msb, lsb)
	Pixel 3	Bits 1, 5 (msb, lsb)
	Pixel 4 (right)	Bits 0, 4 (msb, lsb)

Mode 2: Bit mapped as normal - leftmost pixel msb

(lsb = least significant bit; msb = most significant bit)

Mode 2 is straightforward enough, each bit of a byte represents one pixel on the screen: if the bit is set, the pixel will be illuminated in the selected 'ink' colour, otherwise it will be set to the current 'paper' colour.

However, Modes 0 and 1 are not so easy to deal with, since each pixel is encoded as two or four bits respectively, and the bits are not grouped together, but 'interlace' with the bits relating to adjacent pixels. Obviously, two bits are required to encode the four colours available in mode 1, and four bits are required for the 16 colours available in mode 0.

The most efficient answer to this problem is to employ a 'mask', in order to access just one pixel at a time. A pixel 'mask' is a byte in which only the bits relating to a particular pixel are set; all other bits are reset. The masks required for modes 0 and 1 are as follows:

MODE 0:	Pixel 1 (left-hand)	- BIN 10101010, or AA HEX
	Pixel 2 (right-hand)	- BIN 01010101, or 55 HEX
MODE 1:	Pixel 1 (left-most)	- BIN 10001000, or 88 HEX
	Pixel 2	- BIN 01000100, or 44 HEX
	Pixel 3	- BIN 00100010, or 22 HEX
	Pixel 4 (right-most)	- BIN 00010001, or 11 HEX

So in order to access (for example) the left-hand pixel of a particular byte in mode 0, the following routine could be used:

```
LD    A,(HL)    ;HL=address of screen byte
AND   AAH       ;mask off right-hand pixel
```

The accumulator now holds the encoded data for only the selected pixel; all other bits are reset.

Pixel Scroll

As a demonstration of this technique in action, here is a useful routine to scroll the whole screen to the left by one pixel: (the mask is selected for mode 1, but the routine will work in mode 0 by selecting the appropriate mask.)

```
PIXTR     DEFB     0
START     LD       HL, C000H      ; Address of top left corner of screen
          LD       E, 88H         ; mask for left-most pixel in byte
          LD       B, 200         ; Number of pixel lines in screen
LOOP 1    PUSH     BC
          PUSH     HL             ; save address of first byte on stack
          LD       A, (HL)        ; get first screen byte
          AND      E              ; mask off all pixels except left-most
          LD       (PIXSTR),A     ; store pixel for later
          CPL
          AND(HL)                 ; blank out pixel
          LD       (HL),A         ; update screen
          RLC      (HL)           ; rotate screen byte left
          LD       B, 79          ; number of bytes in line minus one
LOOP 2    INC      HL             ; next byte
          LD       A, (HL)
          AND      E              ; mask off all pixels except left-most
          LD       D,A            ; save pixel for later
          CPL
          AND      (HL)           ; blank out pixel
```

```
         LD      (HL), A         ; update screen
         RLC     (HL)            ; rotate screen byte left
         LD      A,D             ; recover pixel
LOOP 3   RRC     E               ; rotate mask right one bit
         JR      C, OUT 1        ; jump out when mas bit rotates out
         SRL     A               ; else shift pixel right
         JR      LOOP 3          ; and repeat
OUT 1    DEC     HL              ; back to previous byte
         OR      (HL)
         LD      (HL), A         ; insert pixel into screen byte
         INC     HL              ; restore screen address
         DJNZ    LOOP 2          ; jump back unless finished with line
         LD      A,(PIXSTR)      ; recall stored pixel
LOOP 4   RRC     E               ; rotate mask right one bit
         JR      C, OUT 2        ; jump out when mask bit rotates out
         SRL     A               ; else shift pixel right
         JR      LOOP 4          ; and repeat
OUT 2    OR      (HL)
         LD      (HL),A          ; insert pixel into screen byte
         POP     HL
         LD      BC, 0800H
         ADD     HL,BC           ; next line down
         JR      NC, END         ; jump if total not greater than FFFFH
         AND     A
         LD      BC,3FB0H
         SBC     HL,BC           ; else subtract 3FB0H
END      POP     BC
         DJNZ    LOOP 1          ; jump back unless finished
         RET
```

This routine can be easily modified to scroll the screen to the right by one pixel, as follows:

```
START    LD      HL,C04FH        ;address of top right corner of scrn.
         LD      E,11H           ;mask for right-most pixel in byte
```

The remainder of the routine will be the same as before, except:

Change all RLC (HL) instructions to RRC (HL)
Change all INC HL instructions to DEC HL
Change all DEC HL instructions to INC HL
Change all RRC E instructions to RLC E
Change all SRL A instructions to SLA A

The screen could be scrolled in mode 2 by the above routines, if the initial mask was set to 80H; however, it can be scrolled much more quickly by a simpler routine, as each pixel is represented by one bit in this mode.

MODE 2 PIXEL SCROLL (RIGHT)

```
START    LD      HL,C000H        ;top left corner of screen address
         LD      B,200           ;number of pixel lines
LOOP 1   PUSH    BC
         PUSH    HL              ;save line start address on stack
         XOR     A               ;zero accumulator and clear carry flag
         LD      B,80            ;number of bytes in line
LOOP 2   RR (HL)                 ;rotate right: carry to D7:D0 to arry
         INC     HL              ;next byte
         DJNZ    LOOP 2          ;loop until finished with line
         POP     HL              ;recover address of first byte in line
         RRA                     ;rotate last bit from carry into A
         OR      (HL)
         LD      (HL),A          ;insert pixel into first byte
         LD      BC,0800H
         ADD     HL,BC           ;next pixel line
         JR      NC,END          ;jump if total not greater than FFFFH
         AND     A
         LD      BC,3FB0H
         SBC     HL,BC           ;else subtract 3FB0H
END      POP     BC
         DJNZ LOOP 1             ;loop unless finished
         RET
```

To convert for scroll left:

```
        LD      HL, C04FH       ; top line R.H. end address
```

Change `RR (HL)` to `RL (HL)`
Change `INC HL` to `DEC HL`
Change `RRA` to `RLA`

Firmware Routines

A large number of the CPC464 firmware routines are available to the uer, via the Firmware Jumpblock. This the area of RAM above BB00H, which is initialised on start-up with the addresses of the various firmware routines in the lower ROM.

The user should always access the firmware via these jumpblocks, as they automtically take care of ROM paging, system variables etc.

In order to access a firmware routine, simply load registers as required by the entry conditions, then CALL the appropriate jump block entry.

There is insufficient space for a full listing of the firmware routines here, but I have listed some of the most useful screen handling routines below:

ROUTINE	ADDRESS	ENTRY CONDITIONS	EXIT CONDITIONS
TXT OUTPUT	BB5AH	A holds char. code	-
TXT SET COLUMN	BB6FH	A holds column no.	-
TXT SET ROW	BB72H	A holds row no.	-
TXT SET PEN	BB90H	A holds pen no.	-
TXT SET PAPER	BB96H	A holds paper no.	-
TXT STR SELECT	BBB4H	A holds stream no.	-
GRA MOVE ABS.	BBC0H	DE holds X co-ord	-
		HL holds Y co-ord	-
GRA SET PEN	BBDEH	A holds pen no.	-
GRA PLOT ABS	BBEAH	DE holds X co-ord	-
		HL holds Y co-ord	-
GRA TEST ABS	BBF0H	DE holds X co-ord	A holds ink col.
		HL holds Y co-ord	-
SCR SET MODE	BC0EH	A holds reqd mode	A holds scn mode
SCR GET MODE	BC11H	-	-
SCR CLEAR	BC14H	-	HL = scrn address
SCR DOT POSN.	BC1DH	* DE holds X co-ord	C = mask for pixel
		* HL holds Y co-ord	A = pixels/byte -1
			-
SCR SET BORDER	BC38H	B holds 1st colour	-
		C holds 2nd colour	

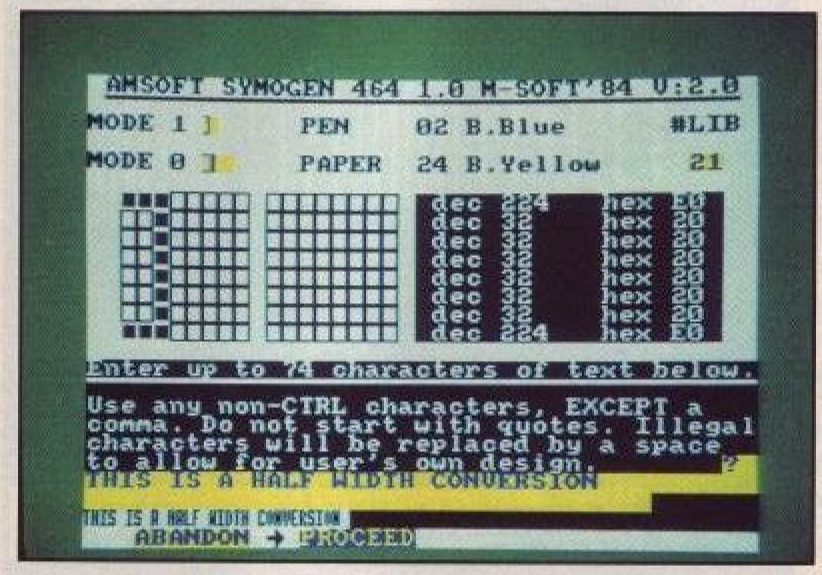
* Co-ordinates required as 'base' co-ordinates, i.e. measured in pixels, and therefore dependent on current screen mode.

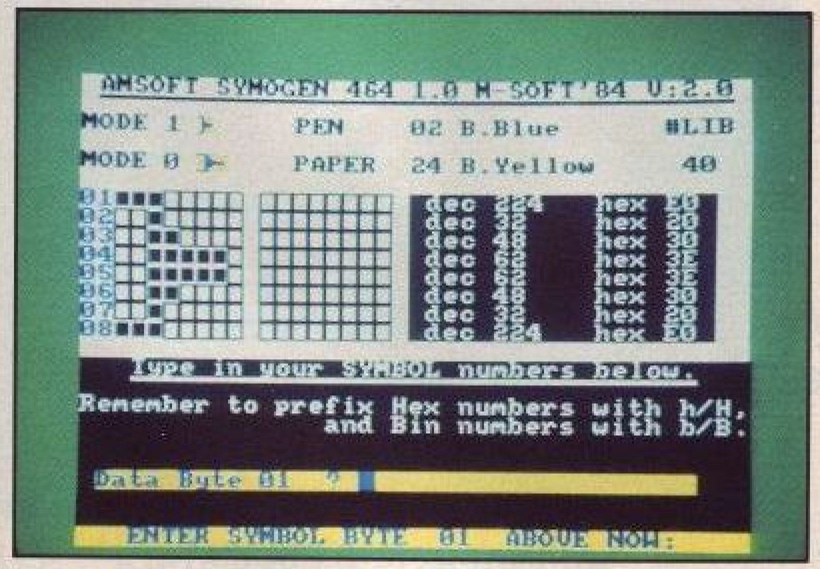
The Firmware Manual SOFT158 for the CPC464 is available from AMSOFT, and is an essential aid for the serious assembly language programmer, as it describes the firmware in detail, and explains the organisation of the hardware interfaces in relation to the firmware.

CPC464 USER CLUB

£2,000 Software Competition

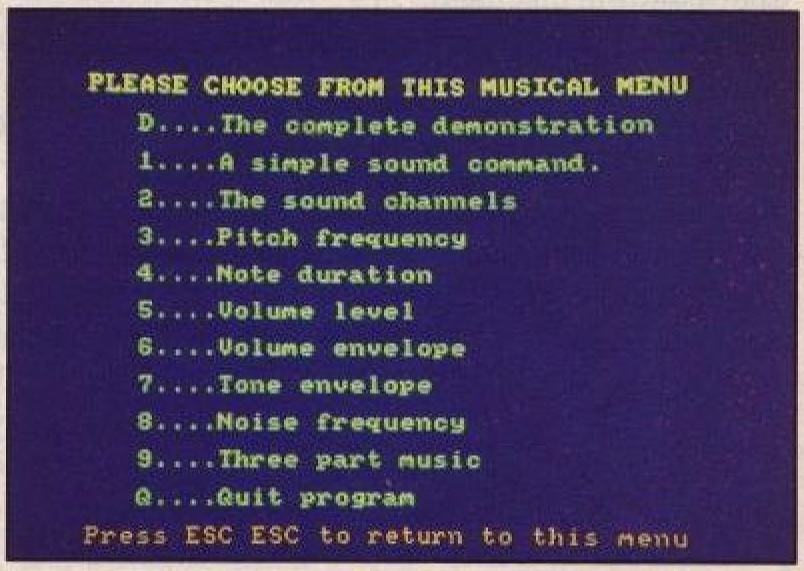
We can't all be winners so it is no surprise that the standard of even the runners-up in the £2,000 user club competition was high. The winner was Mr P.R. Murfitt who regular readers will remember entered a character modifier for the second competition. It was an improved version of this, called 'Symgen' which won the big prize. The documentation had been expanded to include examples and screen shots. The input was very user friendly and the program used the machine well.





SYMGEN - P.R. MURFITT

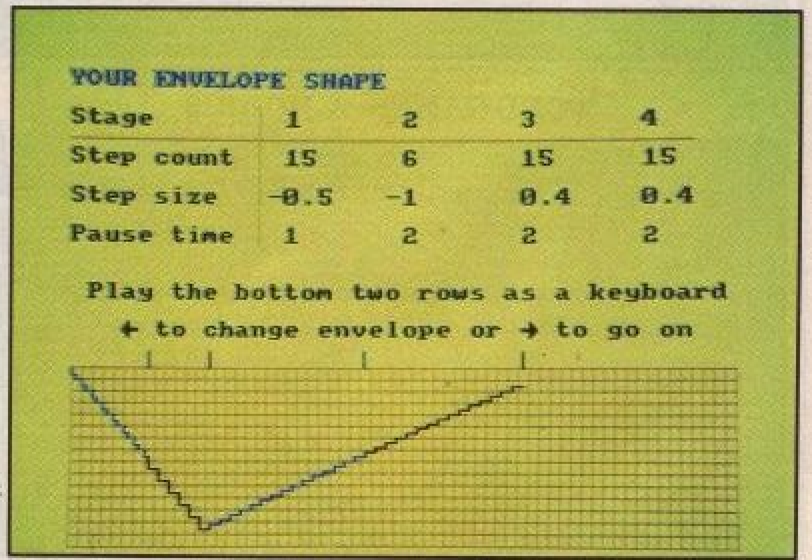
A close runner up was Jonathan Stidwill with his music composer program. This provided facilities for editing volume and tone envelopes, it was more of a tutorial than a music composition program and explained how the computers sound command worked. The graphic representation of the sounds made it easy to work out what was going on. It was the clear approach to the subject which appealed to the judges. An educational entry which was quite impressive was 'Word Games' from Mr A. Norris. This required the



MUSIC COMPOSER - J. STIDWILL

player to make the words given in a list from a jumbled collection of the appropriate letters. The skill comes from deciding in which order the words should be constructed. The speed and neatness of the program was impressive.

The number of games entered was quite disappointing, Peter Smiths 'Darts' being one of very few. This was quite fun but lacked lasting appeal. Zygolian Invasion from D L Lau was a nice implementation of the old shoot-'em-up type game,



MUSIC COMPOSER - J. STIDWILL

with a bit of work and some machine code it could be a very nice game.

Tutorial programs made a strong showing, in

addition to J. Stidwill's music composer there was a tutorial on the physics of sound from Mr A J White and one on the obscure animals of the world.

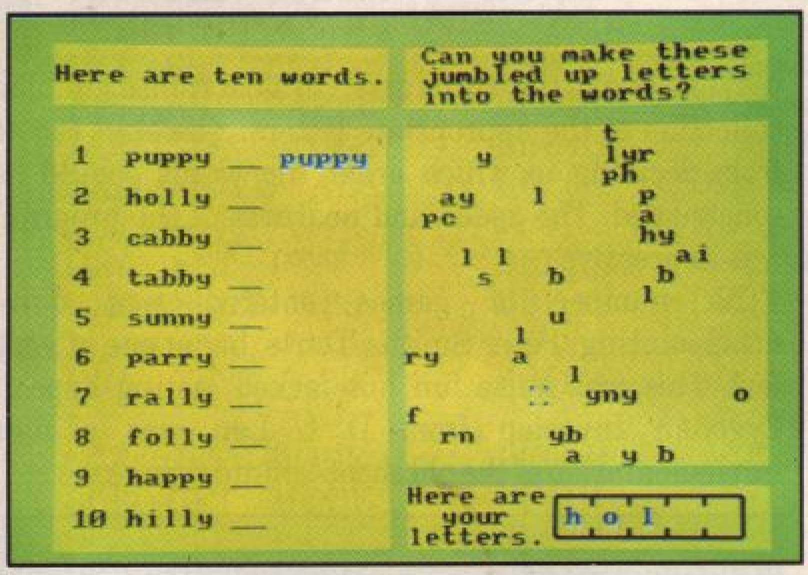
Judging the competition is always fun, however there are a few things which would make our lives a bit easier.

Put your name and address in the first few lines of the program.

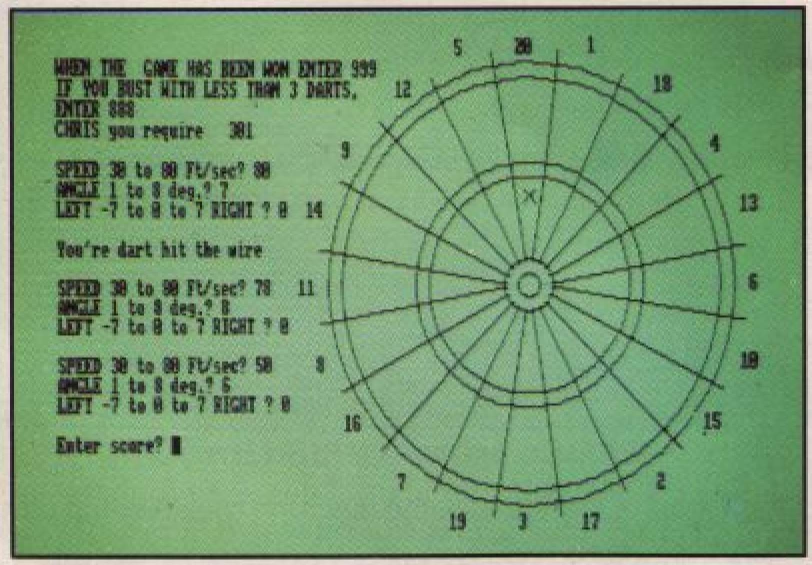
Always write your name and that of the program on the cassette or disc itself.

Try to avoid long title pages, waiting for a tune to play before each game can become very tedious.

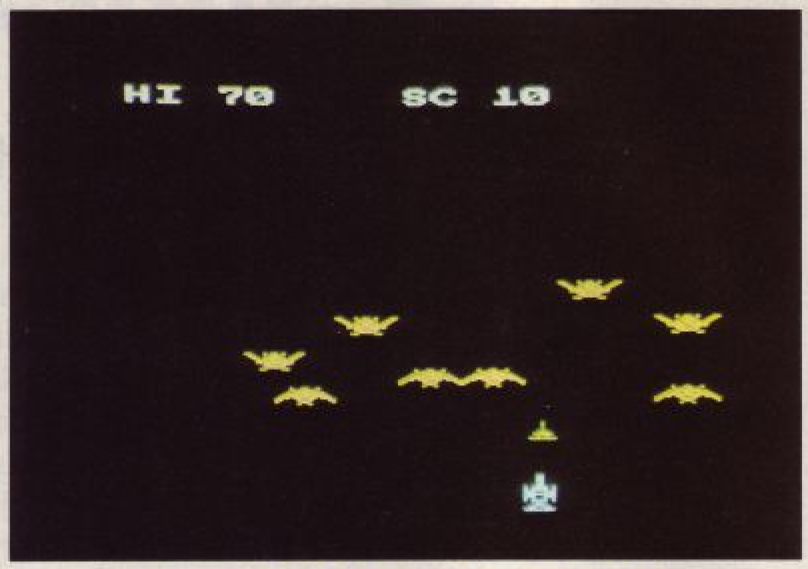
It is becoming harder and harder to pick a winner and we felt this month that it was unfair that such a narrow margin should make the difference between £2,000 and nothing. In future competitions Amsoft may split the prize between several entrants. This might take the form of joint winners of £1,000 each or of a first, second and third of say £1000, £600 and £400. In short the judges will allocate the prize in a way which they deem to be fair. There will still be a chance to win £2,000 but this system should work better.



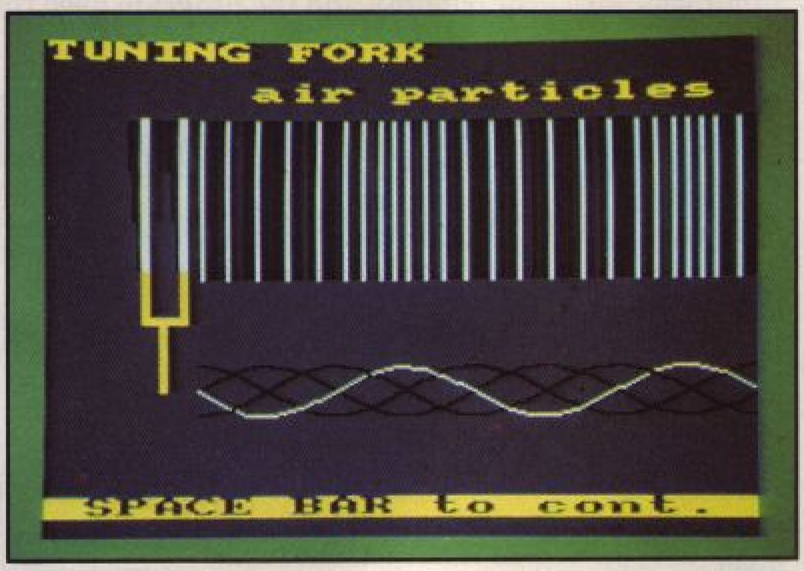
WORD GAMES - A. NORRIS



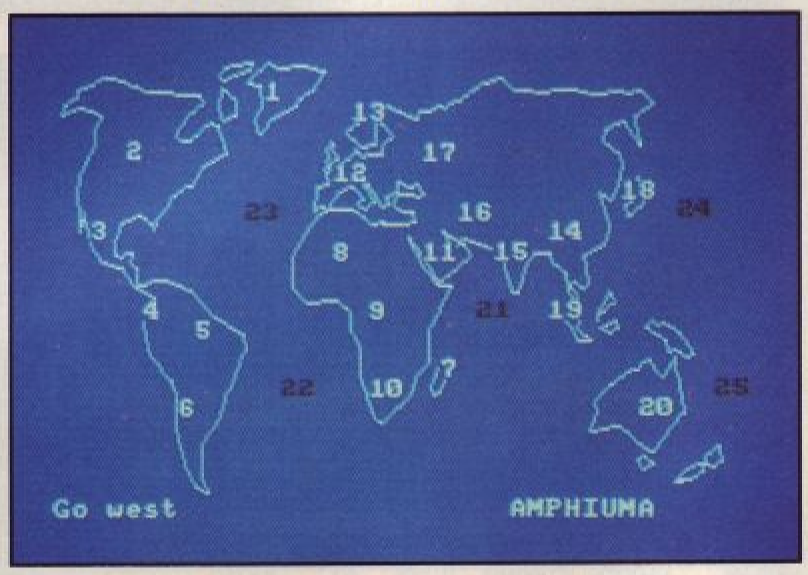
DARTS - P. SMITH



ZYGOLIAN INVASION - D.L. LAU



PHYSICS OF SOUND - A.J. WHITE



ANIMAL QUIZ - R.GREEN.

Since the magazine is going monthly from this issue there will not be a competition in issue 5. You can still send in entries and these will be held over to issue 6. This is because the competition is bi-monthly. We look forward to seeing what you can come up with in the next couple of months, perhaps some entries will arrive on disc. Good luck.

£2,000 program competition!

Think what you could buy with a £2,000 windfall......

The objective couldn't be simpler: write the best program submitted to the AMSTRAD USER review panel. Every other month £2000 will be shared amongst the very best entries.

RULES

1 The winner(s) will be the entrant or entrants who submits the best program. It may be any piece of software that runs on the CPC464.

2 The name of all the winners will be printed in the next issue of AMSTRAD User.

3 All entries must arrive by March 14th 1985. Entries arriving after that date will be included in the next issue's competition.

4 All entries must INCLUDE a disc or cassette copy of the program with some codes where applicable (plus loading instructions where necessary!), AND A SIGNED COMPETITION ENTRY FORM. A full listing would be helpful, but not essential. A brief summary of the program and its purpose should accompany all entries.

5 All entries will be treated in strict confidence.

6 Neither AMSOFT nor Amstrad can be held to be responsible for any loss or damage to any submission.

7 No correspondence can be entered into concerning programs submitted for the cash prize competition.

8 The decision of the judges is final.

9 It is a condition of entry that all entrants have exclusive ownership of the copyright of the material submitted, and that the winners agree to assign all copyright in the winning program to AMSTRAD. All entrants must undertake not to submit the same or a similar program to any other magazine, publisher or organisation until after the announcement of the winning entry.

10 AMSOFT may offer to publish programs either in AMSTRAD USER or as commercial software, in which case amsoft will agree the terms on an individual basis with the author(s) concerned. AMSOFT reserves the right to amend, alter or revise any program that it publishes.

11 No employees of AMSOFT or AMSTRAD, or their relatives may enter this competition.

12 All entries must have the name and address of the programmer as REM's of the comments in the first few lines of the program

HOW TO ENTER:

Think about your program, and map it out in a series of events or features. Write the program onto cassette, based around these events and check that the program runs as intended. Once you are satisfied, send a copy of the cassette in a suitable envelope along with the following:

1) A brief summary of the program in 500 words or less.
2) The completed competition entry form, which must be signed.
3) Program listings, if available.
4) SAE if want your entry returned.

SOFTWARE *COMPETITION*

AMSTRAD CPC464 USER

No entry is valid unless accompanied by this official entry form.
Only User Club members are eligible to enter.

User Club Membership Number ....
Name ....
Address ....
....
Postcode ....
Telephone number ....

***NB All programs must be submitted in UNprotected form. Preferably on disc!**

I have read the rules published in AMSTRAD CPC464 USER, and agree to be bound them.

I have enclosed a:

- ☐ tape/disc
- ☐ listing
- ☐ SAE for the return of my entry
- ☐ A summary of up to 500 words

I am over 18 years of age. YES/NO : Age if under 18 ......

(The signature of a parent or guardian is required where the entrant or the nominated representative of the entrants is under 18 years of age.)

SIGNATURE .... Date ....
Program name ....

THINKING LOGICALLY

Getting the most from rarely used functions by K. Hook.

An often neglected feature of the computer is its ability to perform logicial operations The blame does not always rest with the programmer, more often than not, it is the poor implementation of boolean functions by the computer's ROM. With the CPC 464, however, there is no excuse: Amstrad Basic is very powerful and allows a full set of logical operations to be performed in basic statements.

In boolean algebra there can only be two answers; FALSE [0], or TRUE [1]. If the result of a logical operation is false the computer sets the byte to all zeros. If the result is true then the 8 bits are all set to ones.

Whenever you program the computer with an IF, THEN, and ELSE statement you are actually using logical expressions:

```
10 IF X = Y1 and Y = 2 THEN GOTO 30 ELSE
   GOTO 100
```

IF X = Y1 and Y = 2 then the condition is true, and the program will jump to line 30. Any other values in X and Y will result in the program jumping to line 100. Logical expressions in the form: IF G THEN GOTO 100 can also be used on the CPC 464. Whenever G <> 0 the program will always jump to line 100 -this is a very useful shorthand device which saves a lot of typing.

The use of AND, OR, and NOT is not restricted to simple relational expressions as mentioned above. AND, OR, and NOT can also be used for boolean operations, bit manipulations, and bit comparisons. At this point it is worth examining how each operation functions on a given number.

NOT

The NOT expression forms the 2's complement of the number by inverting each bit and adding one.

```
Formula => NOT A      =-(A+1)
NOT 1   => result returned = -2
NOT -2  => result returned = 1
NOT 0   => result returned = -1
```

AND

The AND operation is used to mask out certain bits of a number.

```
12 AND 4   => result returned = 4
25 AND 12  => result returned = 12
4  AND 2   => result returned = 0
```

Note: This is because 4 = 0100 binary, and 2 = 0010 binary. When these two values are ANDed the result is zero.

OR

This operation is frequently used to set certain bits without affecting the other bits of the byte(s).

```
 4 OR  2 => result returned = 6
-1 OR -2 => result returned = -1

                 11111111 = -1
                 11111110 = -2
                 11111111 = -1
                 ------------
      when OR'd = 11111111 = -1
```

If you have never worked with binary numbers, or do not know how the computer stores its numbers, type in Listing One. Not only is it a good example of how logical operations can be used within a program, but it will also help you convert decimal to binary, and you can see at a glance how each bit alters for different values.

Before going any further, it would be prudent to mention that 1 byte = 8 bits, and by convention the bits are numbered LSB (Least Significant Bit) = 0 (which is the right hand bit) and MSB (Most Significant Bit) = 7. Each bit position represents a power of two which makes conversion from binary to decimal a simple affair.

Now, what does all this mean? By using logical operations we scan set, re-set, and test bits within a byte direct from Basic - machine code programmers will appreciate the significance of the last statement: programs will run faster, programming space will be shortened, and eight 'yes'/'no' status flags can be contained within a byte.

The following formulas will allow you to test for any eventuality within a byte without leaving basic, e.g. the JOY(0) returns with an integer value depending on which movement has been made by the operator. If the joystick has been moved to the left, the value returned will be 4. If the operator pressed the fire button at the same time, the value returned by JOY(0) will be 20. It is a simple matter, using the

information listed below, to write a sub-routine that checks the status of each bit.

Note the CPC464 stores integers in two contiguous bytes which allows a number in the range -32767 through to 32767 to be stored in two bytes of memory. The following routines will only work on integers. You must ensure that the variables used within the statements are DEFINTed at the beginning of your program, or you use the format A%.

Formulas

To set any bit X in a two byte integer NM

```
              NM=NMor2↑X
To set bit 9: NM=NMor2↑9
```

To test any bit X in a 2 byte integer NM

```
               IF NM AND 2↑X
                 THEN...ELSE...
To test bit 4: IF NM AND 2↑4 THEN
                 GOTO 1200 ELSE GOTO 50
```

To re-set any bit X in a 2 byte integer NM

```
                 NM = NM AND NOT 2↑X
To re-set bit 3: NM = NM AND NOT 2↑3
```

You an also use the same operations on $trings. CHR$ (n) is a one byte character. By setting up a $tring of length n, you can have as many status flags as you like.

To set bit X in a one byte $tring T$

```
T$ = CHR$(ASC(T$)OR 2↑X)
```

To re-set bit X in a one byte $tring T$

```
T$ = CHR$(ASC(T$)AND NOT 2↑)
```

To test bit X in a one byte string T$

```
IF ASC(T$) AND 2↑X THEN...ELSE...
```

To set up more than eight conditional tests you first need to calculate how many conditions you require, and then create a $tring to the length of that value.

```
$tring length (n) = INT(number of
  conditions/8) +1
```

You can then set up the $tring by:

```
T$ = STRING$ (n,0)
```

Which will create the desired $tring and initialise each bit to the 'off' condition [0].

You could even set up your one byte $trings in a $tring array. This would make it possible to create a pattern of more than one bit and then test for the pattern throughout the array. For example:

You have a 6 element array of one bytes $trings, and you need to test if any of the $trings have bits 1 & 8 set, you could use the following subutine.

```
100 BTEST$ = CHR$(2↑1) OR (2↑8)
110 FOR COUNT = 0 TO 5
120 IF STRING ARRAY(COUNT) = T$ THEN
    MATCH = COUNT:RETURN
130 NEXT
140 RETURN
```

Variable MATCH will hold the number of the $tring element that matches the test $tring BTEST$. Using arrays will allow you to set up as many one byte $trings as you require.

All the above formulas can be further enhanced by programming them into DEF FN functions. After a little thought you will find many uses for the above routines, and finally, consider the following:

```
X=X+2: IF X> 3046 THEN X=3046
```

IS EXACTLY THE SAME AS =>

```
X=X-(X+2<3046)
```

LISTING ONE:

```
1  MODE 1:PAPER 0:CLS ' BOOLEAN
    BIT SETTING
10 WINDOW #1,8,30,3,3
20 WINDOW #2,8,34,5,5
30 WINDOW #3,8,34,7,7
40 WINDOW #4,8,34,8,8
50 LOCATE #1,1,1
60 CLS#1:INPUT#1,"INPUT YOUR
    NUMBER";A
70 LOCATE #2,1,1
80 PEN#2,2:INK 2,5:PRINT#2,
    "YOUR NUMBER = ";A
90 LOCATE #3,1,1:PEN#3,5:INK 5,24
100 PRINT #3,"BIT  0 1 2 3 4 5 6 7 "
110 LOCATE #4,6,1:PEN#4,6
120 FOR I=0 TO 7
130 IF A AND 2↑I THEN
    PRINT#4,"1 "; ELSE PRINT#4,"0 ";
140 NEXT
150 GOTO 50
```

LARGEST COMPUTER CENTRE
IN
MANCHESTER

MAIN AMSTRAD STOCKIST
+
ALL LEADING COMPUTERS

Executive range of AMSTRAD software, books and addons.
Personal export scheme. Export dealer enquiries welcome
24 hour. U.K. mail order facility.

MIGHTY MICRO
SHERWOOD CENTER 268 WILMSLOW ROAD
FALLOWFILED, MANCHESTER TEL: 067-224 8117

Computer World UK Ltd of BOLTON
were proud to be the very FIRST outlet for Amstrad computers in the North West.

Now, in 1985, we are still proud to be offering the widest and strongest support for the CPC464.

Printers: Disc Drives: Software: Magazines: Books: Speech Synthesisers.

FIRST, & STILL THE BEST:
COMPUTER WORLD (UK) LTD

208 Chorley Old Road
Bolton Lancs
BL1 3BG

Open 10.30-7.30 Mon-Fri
9.00-5.00 Sat

Tel (0204),494 304

FOR AMSTRAD, MSX, DRAGON ETC.
BUSINESS ACCOUNTS
PROFESSIONAL STANDARD PROGRAMS FOR:
STOCKRECORDING, INVOICING,
SALES, PURCHASE & NOMINAL LEDGERS(WITH VAT)

CASSETTE BASED WITH AUDIT TRIALS FOR MANUAL INTEGRATION
CRASH PROOFED' AND VERY USER FRIENDLY WITH
STEP-BY STEP MANUALS REQUIRING.NO COMPUTER KNOWLEDGE

COMPLETE SYSTEMS (all hardware and software) from **£449+VAT**
(PROGRAMS ONLY AT £25+EACH OR ALL 5 FOR £99+VAT)

DESIGNED FOR SMALL/ONE MAN BUSINESS TO PROVE AT MINIMAL COST THAT 'THE WIFE' CAN COPE WITH A COMPUTER AND THAT IT WILL HANDLE YOUR BUSINESS ACCOUNTS, BUT PROVIDING AN UPGRADE PATH TO DISC-BASED/INTEGRATED SYSTEMS USING THE SAME PROFESIONAL QUALITY PRINTER AND REQUIRING VIRTUALLY NO OPERATOR RE-TRAINING IF VERY FAST OPERATION IS REQUIRED.

SEND LARGE STAMPED ENVELOPE FOR FULL DETAILS
TELEPHONE SUPPORT AVAILABLE AT EXTRA COST
DEALERS ENQUIRES WELCOME

A **10-GAME** COMPENDIUM FOR AMSTRAD OR MSX - SEND JUST £12

COMPUTER CONTACT (SALES)
50 HEADINGLEY RD., RUSHDEN, NORTHANTS tel. (0933) 55673 or 56894
PRICES MAY CHANGE WITHOUT NOTICE - OFFERS ARE SUBJECT TO AVAILABILITY

MIDLAND COMPUTER LIBRARY
ATTN: AMSTRAD OWNERS
The First & Largest AMSTRAD Library in the World
(often copied but never equalled)

1 All the latest titles on cassette and disc
2 Your first tape hired free
3 Life membership £6
4 Hire charge from only £1.50 inc p&p for full 7 days
5 No limit to the amount of games you can hire
6 All games originals with full documentation
7 Games, Business and Educational software
8 Return of post service
9 Lowest new hard and software prices in the UK
10 Join now on 14 day money back guarantee
11 Over 200 games in stock

MIDLAND COMPUTER LIBRARY
172 Studdley Road Greenlands
Redditch B98 7NA Telephone 0527 23584

AMSTRAD SOFTWARE

Jet Set Willy	£8.00	Manic Miner	£7.50
Hunchback	£7.50	Blagger	£7.00
Codename Mat	£7.50	American Football	£8.50
Steve Davis Snooker	£6.40	Roland in Time	£7.50
Roland Goes Digging	£7.50	Roland on the Run	£7.50
Spannerman	£7.50	Harrier Attack	£7.50
Crazy Golf	£7.50	Amsgolf	£7.50
Hunter Killer	£7.50	Master Chess	£7.50
Snowball	£8.35	Adventure Quest	£8.35
Lords of Time	£8.35	Dungeon Adventure	£8.35
Gems of Stradus	£7.50	Alien Break In	£7.50
Guide to Basic Pt 1	£17.95	Pascal	£32.95
Devpac	£22.95	Home Budget	£17.95

All categories catered for:
Arcade, Adventure, Strategy, Utilities, Home/Business, Educational
Write or phone for free catalogue

Cheques/P.O. to:
MICRO COMPUTER WORLD
1 LANE CLOSE, LONDON NW2 6QZ. Tel: 01-452-0893
Access taken

ESTUARY
PERSONAL COMPUTERS

Your local AMSTRAD SPECIALIST
We probably have the largest selection of accessories
★ BOOKS ★ SOFTWARE ★ in Essex
VICTORIA CIRCUS SHOPPING CENTRE
(up escalator from Victoria railway station)
★TEL: Southend (0702) **614131**

ARNOLD BLACKWOOD - ADVENTURER EXTRAORDINAIRE

THE TRIAL OF ARNOLD BLACKWOOD
A very worthwhile addition to the Adventurers' library CPC464 USER Oct' 84

ARNOLD GOES TO SOMEWHERE ELSE!
"I found it all immensely lively and enjoyable"....T. Kendle PCW Dec' 84

THE WISE & FOOL OF ARNOLD BLACKWOOD
"Buffs will enjoy the paranoia that pervades the adventure" PCW Dec' 84

NEW ANGELIQUE: THE GRIEF ENCOUNTER
The first Romantic Adventure with an emphatically feminine heroine,

Each Adventure **£6.50** inc. By return post. Chqs. PO's to:

NEMESIS
10 CARLOW ROAD, RINGSTEAD,
KETTERING, NORTHANTS, NN14 4DW

VIDEO TECHNIQUES
BROMSGROVE VIDEO CENTRE LTD

1 CHURCH STREET
BROMSGROVE
WORCESTER
0527 35688

COMPUTER WITH COLOUR MONITOR. £349.00

GREEN SCREEN £239.00

WE STOCK THE HARDWARE AND SOFTWARE FOR THE AMSTRAD CPC 464, SO COME AND SEE US NOW.

Tel: 0527 35688

FORTH

As a prelude to the release of multi-FORTH 83 we are selling a fig-FORTH for the CPC464 on cassette tape. Purchasers will be able to trade in their tapes when multi-FORTH 83 becomes available.

Write or Telephone for more details to:

SKYWAVE SOFTWARE

73 Curzon Road
Bournemouth
BH1 4PW
Telephone 0202-302385

Access VISA

CUT LOADING TIME DRAMATICALLY AND LIST YOUR WELCOME TAPE WITH OUR WELL KNOWN .... " SYCLONE" PROGRAM

SEE THE IMPRESSIVE FEATURES OFFERED BY SYCLONE AND OUR OTHER VERSATILE UTILITES

(Buy any 2 titles and get a 3rd cassette containing a **Real Time Digital Alarm Clock, RSX Program FREE.)**

ZEDIS A comprehensive yet friendly editor and disassembler, indispensible to the novice or Machine Code expert.

- Full Z80 disassembly
- Continual menu display
- Easy to use, full error trapping
- 3 windows and 80 column text
- High speed Hex/Code/String search
- Forward/Backward paging and scrolling
- 3 O/P styles
- Hex/Code/String I/Ps
- Full control of O/P to printer
- Calc. relative displacement & jumps
- Register inspection
- Break point insertion
- Full written instructions to dis. ROMs

CASSETTE £6.95 inc. P/P

RSX.SYCLONE A resident system extention which adds 3 new commands, offering 19 new facilities for your CPC 464.

- Commands available from Basic
- Load and list protected Basic programs
- Copies all tested software
- All file types catered for
- Choice of 4 loading speeds, 1000 up to 4000 baud
- Save your programs to load in up to 1/4 of normal time
- Convert existing software to high speed loading software
- Full error reporting
- Antisyclone facility to add to your own programs
- Full written instructions

CASSETTE £6.95 inc. P/P

PRINTER PAC 1 A resident system extention which can add 6 new commands, a total of 8 extra facilities to your CPC 464 when used with the DMP 1 printer or the Epson RX 80 and similar printers.

Side 1 .... EPDUMP for the Epson RX 80 and similar printers.
Side 2 .... AMDUMP for the DMP 1 printer.

- Specify horizontal position of dump
- Specify which inks you wish to be considered for background
- Dump current screen to printer
- Copy the text only, to printer
- Abbreviated codes to printer. A great time saver when sending more than a few codes
- 3 new type styles for the DMP 1. (Not required for the Epson RX80)

CASSETTE £4.95 inc. P/P

EUROPE - Add £1 per title, REST OF WORLD add £1.50 per title

PRIDE UTILITIES (CP4)
7 Chalton Heights,
Chalton,
Luton,
Beds, LU4 9UF

**** NOW AVAILABLE, OUR GREAT NEW "TAPE TO DISC CONVERSION" PROGRAM-£7.95 INC P/P **** S.A.E. FOR DETAILS.

HOMECOMPUTER
243 Holdenhurst Road

HUGE SELECTION OF
AMSTRAD SOFTWARE
HELPFUL FRIENDLY ADVICE
MACHINES AND PERIPHERALS FROM STOCK
10% DISCOUNT ON SOFTWARE ON PRODUCTION OF THIS ADVERT

Tel. Bournemouth 301691

DCS SOFTWARE

Mail-order software with same day despatch (we mean it!).
You'll find your business, utility, education and games programs listed in the Directory section of this magazine. If you'd like more details, send an SAE for free catalogue.

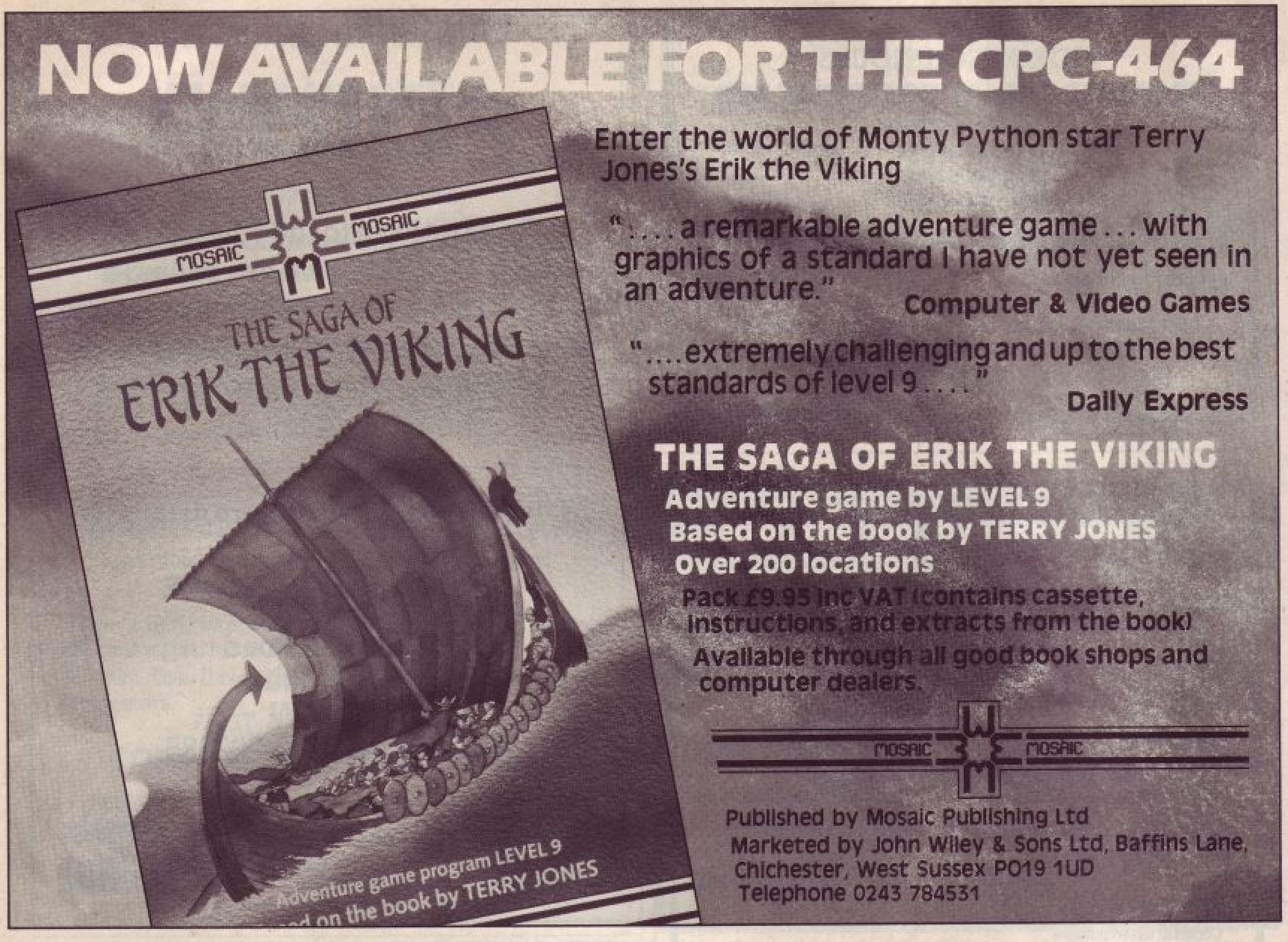
DCS (AM), 38 SOUTH PARADE,
BRAMHALL, STOCKPORT,
SK7 3BJ.

CASSETTE BASED SOFTWARE

Cassette based utility routines with full instructions

Details from, or **£9.95** to:

PROSOL
110 Keswick Road
Lancaster LA1 3LF
(0524) 36474



ADVERTISER'S INDEX

AMSTRAD BBC CBM64 SPECTRUM MEMOTECH NASCOM ATARI

"You really can't go wrong with any Level 9 game as they are all brilliant." *Crash Micro Sept 84*

RETURN TO EDEN

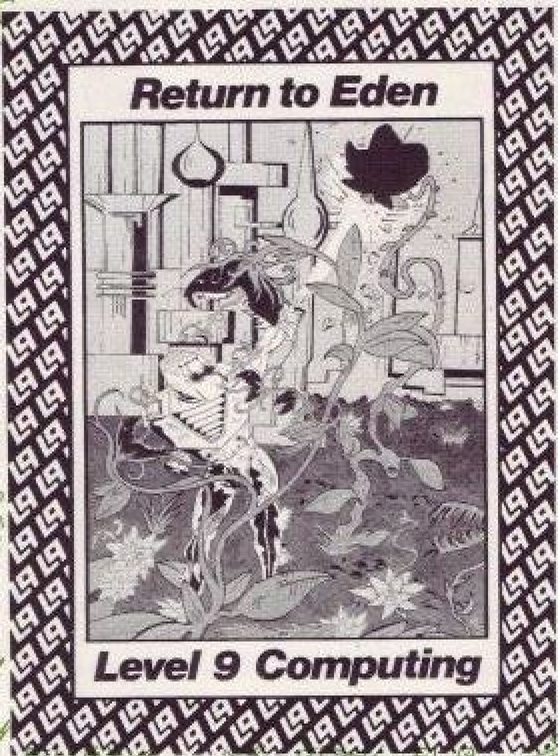
Level 9's first amazing full-colour graphical adventure.

Return to Eden is the long-awaited sequel to Level 9's top-selling Snowball adventure, set on the weirdest planet ever. Now it's here with 240 locations, masses of puns and puzzles and with hundreds of pictures in the AMSTRAD, CBM 64 and Spectrum versions.

"Whichever machine you own, if you have the vaguest tendency towards adventure playing then you must try one of these games (unfortunately you'll probably end up wanting to buy the lot."
– Computing Today, Aug 84

"The Level 9 Adventures are superbly designed and programmed, the content first rate. The implementation of Colossal Cave (Adventure) is nothing short of brilliant; rush out and buy it. While you're at it, buy their others, too. Simply smashing!"
– Your 64, June 84

"Level 9 – arguably the producers of the best adventure games in the UK – have done it again. Lords of Time is a sparkling addition to its stable of winners."
– Acorn User, July 84



"One of the best adventure games I have ever had the pleasure to get my hands on. I can recommend Dungeon Adventure without the slightest fear of being contradicted. This is a massive sojourn into the unknown."
– Memopad, Oct 84

"The Saga of Erik the Viking . . . a remarkable Adventure game. It carries all the hallmarks of a Level 9 Adventure – problem, text display and size of map – with graphics of a standard I have not yet seen before in an Adventure."
– Computer & Video Games, Oct 84

"I thoroughly recommend these Adventures, they are excellent value for money. No self-respecting Adventure-addict should be without them. I believe Level 9 are producing a series of Adventures which should be regarded as classics."
– Page 6, July 84

Available from the HMV Shop and good computer stores everywhere. If your local dealer doesn't stock Level 9 adventures yet, use the coupon to buy them from us, or ask him to contact: Centresoft, Microdealer UK, Lightning, Leisuresoft, R&R, Lime Tree, PCS (SW), MCD, Wonderbridge etc.

	Cassette £9.95	Disk £11.95 BBC / CBM64
COLOSSAL ADVENTURE: The classic mainframe game, with 70 bonus rooms.	☐	☐
ADVENTURE QUEST: An epic puzzle journey through Middle Earth.	☐	☐
DUNGEON ADVENTURE: 40 treasures to find and 100+ puzzles to solve.	☐	☐
SNOWBALL: Immense Science Fiction game with over 7000 locations.	☐	☐
RETURN TO EDEN: Vegetarian adventure. (Amstrad, CBM 64 and Spectrum versions have 240 pictures).	☐	☐
LORDS OF TIME: Imaginative romp through World History.	☐	☐
ERIK THE VIKING: Family game for BBC, CBM 64 and Spectrum only. All with 180+ pictures	☐	

I ENCLOSE A CHEQUE/PO FOR £9.95 PER CASSETTE OR £11.95 PER DISK

My name: __________

My address: __________

My micro is __________
(one of those listed below with at least 32K). Send coupon to:

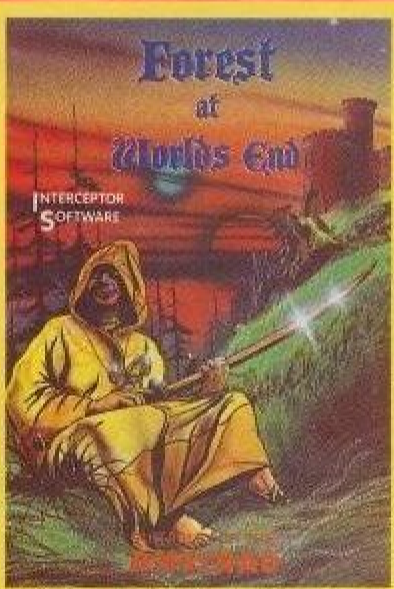
LEVEL 9 COMPUTING
Dept , 229 Hughenden Road
High Wycombe, Bucks. HP13 5PG

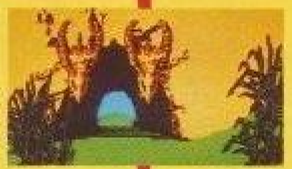
AMSTRAD BBC CBM64 SPECTRUM MEMOTECH NASCOM ATARI

Amstrad Adventures from Interceptor Software

R.R.P. £6.00 EACH

Interceptor Software are proud to present their first 4 releases on the Amstrad. These 'user friendly' graphical adventures not only utilise the memory to a full but display the best graphics yet seen on this new computer. A sensible price of £6.00 each makes them the best value software on the Amstrad.

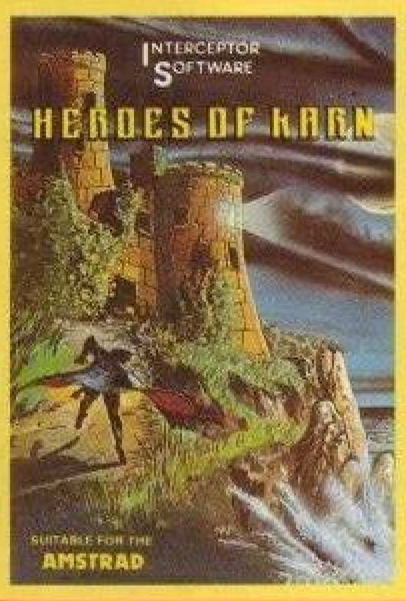


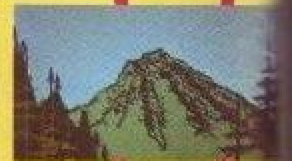


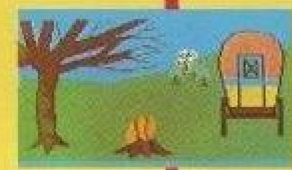


Forest at Worlds End

Forest at Worlds End is a mythical adventure where you have to rescue Princess Mara who has been captured by the evil wizard, Zarn. Many foes await you in the forest!!

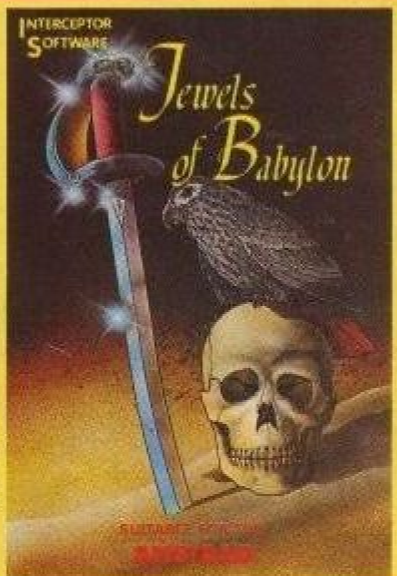




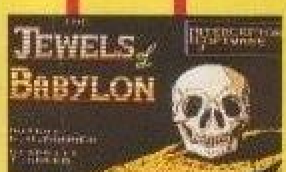


Heroes of Karn

Heroes of Karn the hit adventure on the CBM 64 has now been converted to the Amstrad. This classical story will keep you occupied for many moons and when you finally solve the adventure and find the Heroes you will find a vacuum in your life only to be filled with the follow on, Empire of Karn which will be released in 1985.

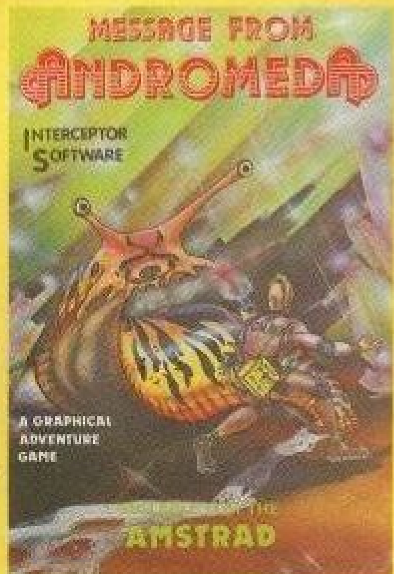


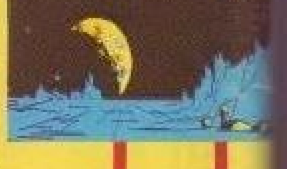




Jewels of Babylon

Jewels of Babylon is set with a pirate theme, where you, the sole survivor of a pirate raid have the task of recovering the Jewels for Queen Victoria who has promised them as a wedding gift to an Indian Princess







Message From Andromeda

Message From Andromeda is a space adventure where you are a captain of a patrol cruiser in the galaxy. The task set for you is laid fraught with danger? Be prepared for the unexpected.

Available from all leading Software Retailers or direct from

INTERCEPTOR SOFTWARE

Lindon House, The Green, Tadley, Hants, England.

Tel:(07356) 71145/3711 Telex: 849101 INMICS G